NUWC-NPT Technical Document 11,138
15 June 1999

Bayesian Classification Using Noninformative Dirichlet Priors

Robert S. Lynch
Surface Undersea Warfare Department

Naval Undersea Warfare Center Division
Newport, Rhode Island

Approved for public release; distribution is unlimited.

PREFACE

This document is an adaptation of the author's 1999 dissertation in electrical and systems engineering for the degree of doctor of philosophy from the University of Connecticut.

Reviewed and Approved: 15 June 1999

Patricia J Dean

Patricia J. Dean
Director, Surface Undersea Warfare

UNITED STATES NAVY

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE 15 June 1999	3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE

Bayesian Classification Using Noninformative Dirichlet Priors

5. FUNDING NUMBERS

6. AUTHOR(S)

Robert S. Lynch

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Undersea Warfare Center Division
1176 Howell Street
Newport, RI 02841-1708

8. PERFORMING ORGANIZATION REPORT NUMBER

TD 11,138

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Adaptation of author's doctoral dissertation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

In this dissertation, the Combined Bayes Test (CBT) and its average probability of error, $P(e)$, are developed. The CBT combines training and test data to infer symbol probabilities where a Dirichlet (completely noninformative) prior is assumed for all classes. Using $P(e)$, several results are shown based on the best quantization complexity, M^* (which is related to the Hughes Phenomenon). For example, it is shown that M^* increases with the training and test data. Also, it is demonstrated that the CBT outperforms a more conventional Maximum Likelihood (ML) based test, and the Kolmogorov-Smirnov Test (KST). With this, the Bayesian Data Reduction Algorithm (BDRA) is developed. The BDRA uses $P(e)$ (conditioned on the training data) and a "greedy" approach for reducing irrelevant features from each class, and its performance is shown to be superior to that of a neural network. From here, the CBT is extended to demonstrate performance when the training data of each class are mislabeled. Performance is shown to degrade when mislabeling exists in the training data, being dependent on the mislabeling probabilities. However, it is also shown that the BRDA can be used to diminish the effect of mislabeling. Further, the BDRA is modified, using two different approaches, to classify test observations when the training data of each class contain missing feature values. In the first approach, each missing feature is assumed to be uniformly distributed over its range of values; in the second approach, the number of discrete levels for each feature is increased by one. Both methods of modeling missing features are shown to perform similarly, and both also outperform a neural network. With these results, the BDRA is applied to three problems of interest in classification. In the first problem, the BDRA is applied to training data containing class-specific features; in the second problem, the BDRA is used to fuse features that have been extracted from independent sonar echoes. Finally, in the third problem, the BDRA is trained and tested on the Australian Credit Card Data (ACCD). In all three cases, the BDRA is shown to improve performance over existing methods.

14. SUBJECT TERMS

Bayesian Theory	Information Theory	Statistics	Probability
Neural Networks	Pattern Calssification		

15. NUMBER OF PAGES
122

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT SAR

TABLE OF CONTENTS

Chapter 1: Introduction **1**

1.1 Problem Statement and Methodology 1

1.2 Previous Related Research 5

1.3 Publications of This Research 12

Chapter 2: The Combined Bayes Test **15**

2.1 Introduction 15

2.2 Combined Information Classification 16

2.2.1 Combined Multinomial Model 16

2.2.2 Combined Bayes Test 17

2.2.3 Probability of Error 19

2.3 Results 22

2.4 Testing the Statistical Similarity of Discrete Data 29

2.5 Summary 32

Chapter 3: The Bayesian Data Reduction Algorithm **33**

3.1 Introduction 33

3.2 Development of the BDRA 34

3.3 Results 39

3.3.1 Performance at Reducing Binary Valued Irrelevant Features 40

3.3.2 Performance at Reducing Ternary Valued Irrelevant Features 48

3.4 Summary 51

Chapter 4: The CBT and Mislabeled Training Data 53

4.1 Introduction 53

4.2 Classification With Mislabeled Training Data 54

4.2.1 Combined Multinomial Model 54

4.2.2 Combined Bayes Test (CBT) 55

4.2.3 Probability of Error 57

4.3 Results 58

4.4 Applying the BDRA to Mislabeled Training Data 62

4.5 Results Using the BDRA 64

4.6 Summary 67

Chapter 5: The BDRA and Missing Features 69

5.1 Introduction 69

5.2 The BDRA Extended for Missing Features 70

5.2.1 Method 1 70

5.2.2 Method 2 74

5.3 Results 74

5.4 Summary 77

Chapter 6: Application of the BDRA to Miscellaneous Problems in Classification 79

6.1 Introduction 79

6.2 The BDRA Applied to the Selection of Class-Specific Features . . 80

6.3 The BDRA Applied to the Fusion of Features From Independent Sonar Echoes 86

6.4 The BDRA Applied to the Australian Credit Card Data 91

6.5 Summary 95

Appendix A: Results Using Empirically Generated Data 97

Appendix B: Development of the Combined Bayes Test 101

Appendix C: Development of $f(\mathrm{y} \mid \mathrm{x}_k, H_k)$ from Probabilistic Considerations 105

Appendix D: Mean and Variance of the Probability of Error for Dirichlet Distributed Symbol Probabilities 107

Bibliography 111

LIST OF TABLES

1 Threshold Settings for Each Feature Before Applying the BDRA . 89

2 Initial Quantization for Each Feature of the ACCD 92

3 Performance of the BDRA and a Neural Network With the ACCD 93

4 Final Quantization for Each Feature of the ACCD 94

LIST OF FIGURES

1 Representative training sets for two classes. 2

2 Conceptual diagram of combined information classification. 7

3 $P(e)$ for various numbers of discrete symbols M. 22

4 $P(e)$ for the CBT and the SGLRT. 24

5 M^* for various training data set sizes. 26

6 Comparison of $P(e)$ using different Dirichlet priors. 27

7 The minimum analytical $P(e)$ for the CBT and the KST. 30

8 Performance of the BDRA with six binary valued features, and twenty five samples of training data for each class. 41

9 Performance comparison of the BDRA to a neural network with binary valued features, and twenty five samples of training data for each class. 42

10 Performance results for the situation shown in Figure 8 with one hundred samples of training data for each class. 45

11 Performance comparison shown in Figure 9 with one hundred samples of training data for each class. 46

12 Estimated number of relevant binary valued features by the BDRA. 48

13 Performance of the BDRA with six ternary valued features, and twenty five samples of training data for each class. 49

14 Performance comparison of the BDRA to a neural network with ternary valued features, and twenty five samples of training data for each class. 50

15 $P(e)$ with various mislabeling probabilities. 59

16 $P(e)$ with more test observations. 61

17 Performance of the BDRA with the mislabeling probabilities of Figure 15. 65

18 Average number of relevant features reduced, out of a total of six, from the training data of each class. 66

19 Performance comparison of the BDRA to a neural network when a random number of missing features occurs with a probability of 0.15. 75

20 Performance comparison of Figure 19 repeated using the BDRA and Method 2. 76

21 Performance of a class-specific classifier with binary valued features, and five samples of training data for each class. 82

22 Performance of Figure 21 repeated with fifty samples of training data for each class. 84

23 Performance of the BDRA with class-specific features when applied to the situation of Figure 21. 85

24 Target recognition performance comparison of the BDRA to the Chi-square statistic, and an OR detector. 90

25 Simulated performance comparison of the CBT, CGLRT, and the SGLRT where $N_{\mathrm{y}} = 2$. 98

26 Simulated performance comparison of the CBT, CGLRT, and the SGLRT where $N_{\mathrm{y}} = 25$. 99

27 Simulated performance comparison of the CBT and the KST. . . 100

Chapter 1

Introduction

1.1 Problem Statement and Methodology

A problem that has received much consideration in the technical literature involves classification when the statistics (probabilistic models) of each class are unknown and determined empirically (many examples of this can be found in [1-67]). However, an aspect of this problem that has received little attention is Bayesian classification of discrete observations given a Dirichlet (completely non-informative) prior is assumed for the symbol probabilities of each class. Therefore, the focus of this dissertation is the performance of this classification method.

By "discrete" it is meant that data used to represent each class can take on one of M possible values. This discrete data may have arisen naturally in its M-level form, or it may have been derived by quantizing feature vectors. For example, three binary valued features can take on $M = 2^3 = 8$ discrete symbols corresponding to the eight feature vectors; $(0,0,0)$, $(0,0,1)$, ..., $(1,1,1)$. In this case, the fineness of proposed quantization is of interest, and an important aspect of this work is to provide guidance on this issue. As it turns out, this is a direct result of the ability to place a uniform prior on discretized feature vectors.

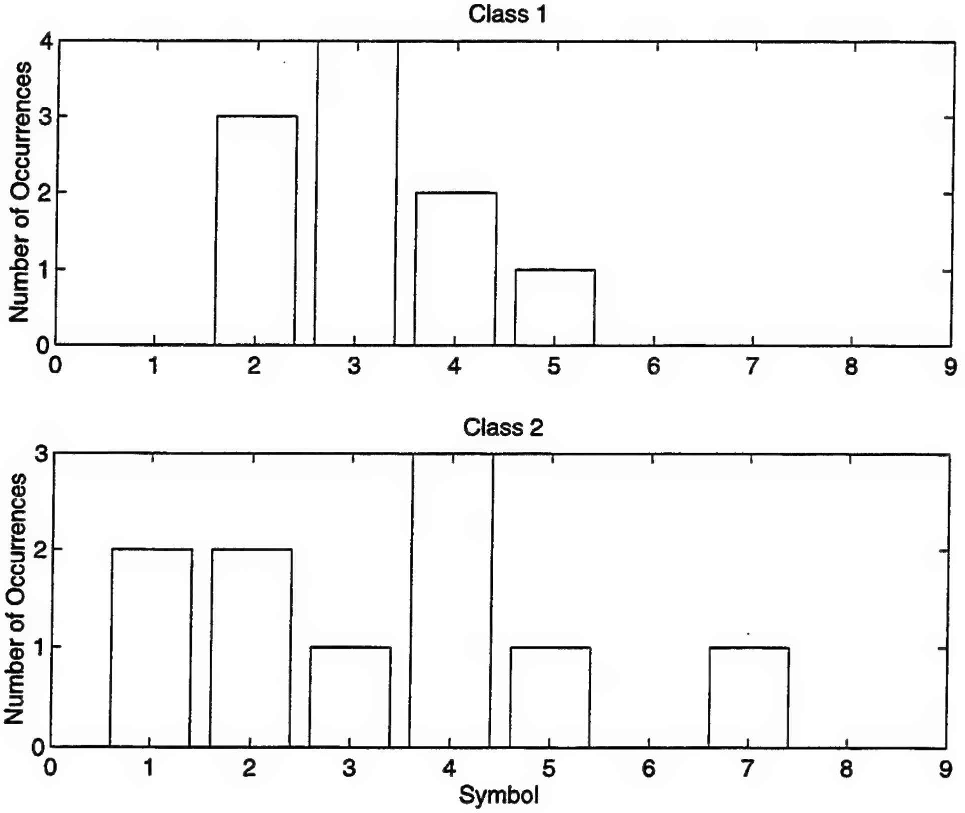


Figure 1: Representative training sets for two classes.

In the situation of interest there are certain labeled realizations of this (M-valued) data, and this is referred to as "training" data under both classes. That is, there are N_k realizations under class k and N_l realizations under class l. As an example, Figure 1 above shows representative histograms of the training data for two hypothetical classes. In this figure, there are ten samples of training data for each class, and eight discrete symbols are assumed. It can be seen that the difference between these two histograms constitutes the relevant classification information for discriminating between them. Now, given this training data, it is expected that N_{y} unlabeled and quantized "test" data are observed, and these are to be simultaneously tested by a classifier. Notice, the goal is to determine, with minimum probability of error, from which class the unknown test data have been generated. Conditioned on the active class all discrete observations of training and test data are assumed independent. Thus, it is reasonable to suppose that the observations are controlled by an underlying multinomial distribution, with the parameters of this distribution – the probabilities of each of the M symbols – unknown and presumably different for each class.

The unknown symbol probabilities for each class can be estimated via maximum likelihood (ML) in the obvious way: the estimate of the probability of the i^{th} symbol is the number of observations of type i in the appropriate training data set, divided by the number of these training data. Using these estimates testing may proceed; performance suffers, however, from singularities caused by test observations being of types unobserved during training. Notice, this type of test has been referred to as the plug in (PI) method, [46, 52], or the maximum frequency recognition rule, [34]. Here, this test is referred to as a standard generalized likelihood ration test (SGLRT), and its performance is discussed in Chapter 2, and Appendix A. In fact, the correctly-posed *generalized* likelihood procedure relies on probability estimates culled from both training and test observations, and here this is what is known as a *combined* test. In this case, based on a combined multinomial distribution the combined generalized likelihood test (CGLRT) appears, and its performance discussed, in Appendix A. But, it is clear that since test data are included in the symbol probability estimates, the problem of an "unrepresented" test symbol is completely avoided.

Thus, the approach of this research could have been based on a combined generalized likelihood framework (i.e., the CGLRT) due to its appeal from a practical perspective. But, from a theoretical standpoint it is less attractive because it lacks optimality in non-asymptotic situations. Therefore, the approach used here is Bayesian: that is, a uniform prior distribution (prior information of complete ignorance) is assumed for the symbol probabilities. Given a prior distribution on all unknown parameters the hypotheses become "simple," and likelihood function based classification is both reasonable and optimal.

Generally speaking, a uniform prior on unknown parameters is common as a basis for testing. But the approach often labors under the necessity that the prior be improper and "diffuse," or even may be uncertain due to the difficulty of expressing uniform-ness in a meaningful way within a complicated parameter space. In this situation, however, a uniform prior is known (and credited to Dirichlet), explicit (but not trivial), and denotes probabilities uniformly-distributed over the positive unit-hyperplane (i.e., an M dimensional space whose elements sum to unity). Further, although training and test data from a common class are statistically-independent given the symbol probabilities, formation of the likelihood function requires integration over of their (product) distribution against the

(Dirichlet) prior and hence expresses their dependence. Such a test, therefore, is combined in its form, and it is referred to in this work as the *Combined Bayes Test* (CBT). A further reason for reliance on this model is that analytic probability of error figures can be calculated, and as will be seen these can be used as a basis for design.

1.2 Previous Related Research

The application of a uniform prior distribution to classifying discrete observations was previously studied by Hughes, [34]. In this work, Hughes showed that for a fixed amount of training data for each class, and a single test observation, the average probability of error is minimum at a certain measurement complexity, M^*. In regards to terminology, Hughes actually showed that the average probability of correct recognition was maximum, but in terms of being a measure of performance quality, this is the same as the minimum probability of error. Also, in addition to being called the the best quantization complexity, M^* is also called the best number of discrete symbols. Additionally, M^* is often referred to as the *Hughes phenomenon* and it has appeared in well known pattern recognition literature such as that by Duda and Hart, [23] (also see Fukanaga, [25]). However, Hughes originally developed his result in terms of the maximum frequency recognition rule (i.e., the SGLRT) instead of the correct Bayesian decision rule that is based on the Dirichlet distribution (this was later pointed out in [1, 14]). But, as it turns out, for the case of a single test observation Hughes' results are valid because the Bayesian decision rule and the SGLRT perform identically.

In the decade following the publication of Hughes' paper other papers appeared suggesting that a best quantization complexity, M^*, does not apply. For example, in [13, 21, 22, 41, 62] it is shown, and for any training set size, that the probability of error approaches zero as the number of independent features approaches infinity. Notice, this represents an apparent contradiction to the results of Hughes which show that once the point M^* is reached the probability of error tends to increase with additional feature information. But, this apparent "paradox" was later resolved in a paper by Van Campenhout, [61]. In this paper, Van Campenhout points out that the quantity M^* is based on incomparable priors, meaning that each value of M represents a different true prior which cannot be compared to another prior. However, an alternative and equally valid interpretation of M^*, and one which is of interest here, is that M^* represents the best combination of discrete symbol quantity and numbers of training data for estimating symbol probabilities (see [23]). Note, the relationship between sample size and estimation ability, as they affect performance, has also been addressed in [26, 32, 39, 57].

A primary contribution contained in this dissertation is development of the CBT (see Chapter 2), which is a generalization of the findings of Hughes to more than one observation of test data. Therefore, Hughes' work is revisited and extended by comparing performance of the CBT to an uncombined SGLRT. In particular, it is shown that larger numbers of test data cause M^* to increase for the CBT with an overall reduction in its average probability of error. However, for the SGLRT larger numbers of test data cause M^* to either remain unchanged or decrease, and its overall average probability of error increases. With these results, it is also shown that with a slight modification the CBT can be used to test the statistical similarity of two discrete data sets (i.e., were they produced by the same multinomial distribution). In this application, the CBT is shown to have a lower average probability of error than the more conventional Kolmogorov-Smirnov Test (KST).

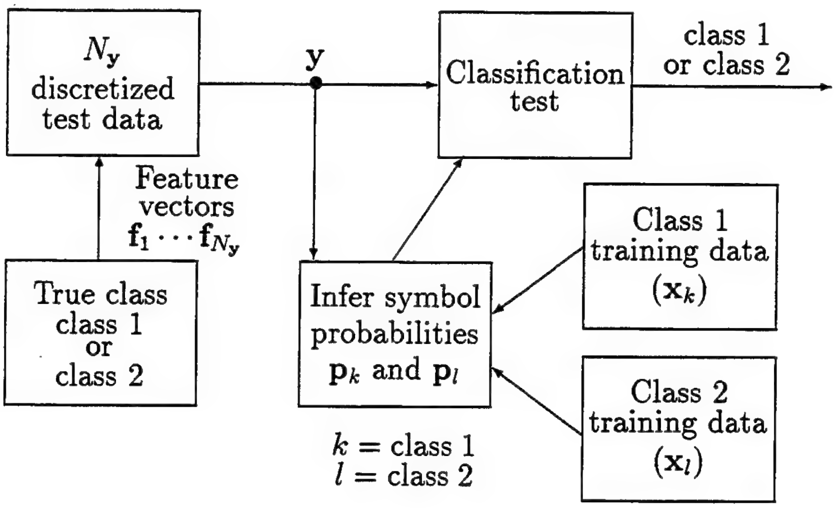

$N_\mathbf{y}$ discretized test data
$\mathbf{y}$
Classification test
class 1 or class 2
Feature vectors $\mathbf{f}_1 \cdots \mathbf{f}_{N_\mathbf{y}}$
Class 1 training data $(\mathbf{x}_k)$
True class class 1 or class 2
Infer symbol probabilities $\mathbf{p}_k$ and $\mathbf{p}_l$
Class 2 training data $(\mathbf{x}_l)$
k = class 1
l = class 2

Figure 2: Conceptual diagram of combined information classification.

In the CBT, both training and test data are combined to infer the true symbol probabilities while, simultaneously, the test data vector is tested for class membership (alternative approaches to simultaneous detection and estimation can be found in [3, 6, 36, 49]). To illustrate this, consider Figure 2 above which shows a conceptual diagram of combined information classification. Notice, by combining all available data the CBT is particularly effective at classifying a target when distributional mismatches exist between the training and test data (for more on this as applied to speech recognition see, [24, 35], also see Appendix A). A likely explanation for this effectiveness is that combining training and test data to infer symbol probabilities implies an adaptation of the test space, [58].

The concept of combining training and test (i.e., labeled and unlabeled) data to improve classification performance has previously been studied by other authors. However, results identical to those shown here have not been found. For example, Merhav and Ephraim, [46] (also see, [24, 48]), discuss empirical results of a method (they refer to this as the approximate Bayesian (AB) decision rule), in which they classify speech signals using Hidden Markov Models. The AB decision rule is based on the joint (i.e., combined) statistics of the training and test data. However, only theoretical results based on large sample size asymptotic situations are provided, whereas the results shown here are presented for any sample size (particular emphasis is placed on small sample sizes). With this, these authors provide a Bayesian decision rule that is credited to Nádas, [52]. The result of Nádas closely resembles the CBT except that as with Hughes' result it is only given for a single observation of test data. Also, in another example, training and test data were combined in [59] (also see [27, 50]) to estimate the parameters of Gaussian mixtures in an application to remote sensing. In this case, it was found that the additional test observations significantly improved overall recognition performance, and an increase in the quantization complexity was also observed.

The Dirichlet distribution (here, the noninformative version of the Dirichlet is used) is the conjugate prior of the multinomial distribution, and in this work it allows the combining of training and test data to infer symbol probabilities. In general, applying the Dirichlet to the multinomial as its prior is well known in the Bayesian statistics literature as the Multinomial-Dirichlet distribution (for example, see [5]). However, in its typical form the Multinomial-Dirichlet is uncombined in that the data are represented by a single random variable. For more on this distribution, many applications of the Multinomial-Dirichlet are found in the general statistics literature (for some examples, see, [8, 16, 19, 28, 29, 51, 60]).

As opposed to a Bayesian approach using the Dirichlet distribution an alternative approach to classifying with discrete training and test data is to use a neural network. It is well known in the literature that a neural network reduces the dimensionality of a training data set by eliminating irrelevant feature information (for example, see [7, 20]). Notice, this has the potential to improve classification performance for a given training data set. Therefore, based on the CBT the Bayesian Data Reduction Algorithm (BDRA) was developed, and later its performance is compared to a neural network (see Chapter 3).

Development of the BDRA represents another contribution of this work, and in the results it is demonstrated to be overall superior to a neural network at reducing a training data set for improved performance. Typically, data reduction is synonymous with feature selection, and in the literature many different approaches can be found (for more on this see, [37, 40] and references therein, also see [2, 9, 18, 23, 25, 43, 54, 63, 65]). However, an aspect of the BDRA not usually found with dimensionality reduction schemes is that it reduces the quantization by reducing (or, in many cases removing) any irrelevant features based on the theoretical Bayes error. Therefore, as opposed to an empirically based method such as adjusting the weights of a neural network and keeping all features, with the BDRA it is easier to see which features are important to correct classification. Additionally, the BDRA has a relatively short training time, and it does not require a randomized starting configuration.

Additional results found in this dissertation involve extending the CBT (and the BDRA) to deal with the problems of mislabeled training data (see Chapter 4), and missing features in the training and test data (see Chapter 5). Each of these problems is treated independently, and in both cases optimal Bayesian tests are developed. Notice, there appears to be little information on these problems by other authors, but, typical of what can be found in the pattern recognition literature is the book by Bishop, [7]. In this book, the severe effects of mislabeled data are briefly discussed and related to estimation in the presence of outliers. Also, with respect to missing features, he describes some of the techniques used to alleviate the problem. For example, a method often used is to 'fill in' missing features by estimates obtained from the known feature values (e.g., such as the sample mean). However, these methods can be prone to problems. With these results, the BDRA is further extended and applied to reducing the training data when it contains class-specific features (see Chapter 6), and most of the related work on this subject can be found in [2]. In this case, the BDRA is shown to be an effective method of selecting ad hoc class-specific features, and it also outperforms the class-specific classifier.

In the literature, the utility of classification methods is often measured by how well they perform with real data, (see the following examples, [9, 24, 26, 31, 32, 35, 46, 54, 59, 63, 65, 66]). Therefore, in addition to the various simulated results appearing in this dissertation two applications are shown in which the BDRA is used with real data (see Chapter 6). The first application involves using the BDRA to fuse features from sonar echoes generated by independent continuous wave (CW) and Frequency Modulated (FM) waveforms. In this case, the sonar echoes were gathered during several at sea experiments, and they typically are used to detect and track surface ships and submarines. Notice, in the literature there does not appear to be another application involving sonar data and an algorithm like the BDRA, and as it turns out, the BDRA performs well at fusing the data for improved target recognition performance. In the second application, the BDRA is used to classify the Australian Credit Card Data (ACCD). The ACCD is based on the actual credit history of 690 applicants, and performance results with other algorithms applied to this data have appeared in [63, 66]. Relative to these other algorithms, the probability of error for the BDRA is comparable to the best that has been achieved with the ACCD.

1.3 Publications of This Research

The following items list all current publications produced by this research including two patent applications.

1. R. Lynch and P. Willett, "Classification With a Combined Information Test," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 1996.

2. R. S. Lynch, Jr. and P. K. Willett, "Classification System and Method Using Combined Information Testing," *Application for US patent*, Navy Case No. 77879.

3. R. Lynch and P. Willett, "Discrete Symbol Quantity and the Minimum Probability of Error for a Combined Information Classification Test," *Proceedings of the* 35th *Annual Allerton Conference on Communication, Control, and Computing*, September 1997.

4. R. S. Lynch, Jr. and P. K. Willett, "Bayesian Classification and Data Driven Quantization Using Dirichlet Priors," *Proceedings of the* 32nd *Annual Conference on Information Sciences and Systems*, March 1998.

5. R. S. Lynch, Jr. and P. K. Willett, "Testing the Statistical Similarity of Discrete Observations Using Dirichlet Priors," *Proceedings of the 1998 IEEE International Symposium on Information Theory*, August 1998.

6. R. S. Lynch, Jr. and P. K. Willett, "Bayesian Classification and the Reduction of Irrelevant Features From Training Data," *Proceedings of the* 37th *IEEE Conference on Decision and Control*, December 1998.

7. R. S. Lynch, Jr. and P. K. Willett, "Bayesian Classification and Discrete Symbol Quantity When the Training Data are Mislabeled," *Proceedings of the 1999 IEEE Information Theory Workshop on Detection, Estimation, Classification and Imaging*, February 1999.

8. R. S. Lynch, Jr. and P. K. Willett, "Classification Using Dirichlet Priors When the Training Data are Mislabeled," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, March 1999.

9. R. S. Lynch, Jr. and P. K. Willett, "Bayesian Classification Using Mislabeled Training Data and a Noninformative Prior," *To appear as an article in a Summer 1999 issue of the Journal of the Franklin Institute.*

10. R. S. Lynch, Jr. and P. K. Willett, "A Data Reduction System for Improving Classifier Performance," *Application for US patent*, Navy Case No. 79550.

11. R. S. Lynch, Jr. and P. K. Willett, "Performance Considerations for a Combined Information Classification Test Using Dirichlet Priors," *To appear as a correspondence in the June 1999 issue of the IEEE Transactions on Signal Processing.*

12. R. S. Lynch, Jr., "Target Detection Performance by Fusing Information from Tracks Generated by Independent Waveforms," *To appear in the Proceedings of the* 2nd *International Conference on Information Fusion.*

13. R. S. Lynch, Jr. and P. K. Willett, "A Bayesian Approach to the Missing Features Problem in Classification," *Submitted for publication in the Proceedings of the* 38th *IEEE Conference on Decision and Control.*

14. R. S. Lynch, Jr. and P. K. Willett, "A Bayesian Approach to Feature Selection Using Noninformative Dirichlet Priors," *Submitted to the IEEE Transactions on Systems, Man, and Cybernetics.*

Chapter 2

The Combined Bayes Test

2.1 Introduction

In this chapter, the Combined Bayes Test (CBT) and a formula for its average probability of error, $P(e)$, are developed. The CBT combines training and test data to infer discrete symbol probabilities where a Dirichlet (completely noninformative) prior is assumed for all classes. Based on the formula for $P(e)$, it is demonstrated that for fixed training and test data set sizes, $P(e)$ reaches a minimum for a particular quantization complexity called M^*. Notice, that the quantity M^* is related to the Hughes Phenomenon of pattern recognition, [34]. With these findings, it is also shown that M^* increases as the training or test data set sizes increase. Further, to show its effectiveness as a classifier $P(e)$ for the CBT is compared to that of the standard generalized likelihood ratio test (SGLRT). While the main body of this chapter is concerned with detailing the classification performance of the CBT, another application of this method is described where it is used to determine if two populations of discrete data are statistically similar. In this case, performance of the CBT is compared to that of the Kolmogorov-Smirnov test (KST).

2.2 Combined Information Classification

2.2.1 Combined Multinomial Model

With this model, it is assumed that there exists a pair of probability vectors, $\mathbf{p}_k$ and $\mathbf{p}_l$, the i^{th} elements of which denote the probability of a symbol of type i being observed under the respective classes k and l. The fundamental model for this testing method is thus formulated based on the number of occurrences of each discrete symbol being an i.i.d. multinomially distributed random variable. Therefore, the joint distribution for the frequency of occurrence of all training and test data with the test data, $\mathbf{y}$, a member of class k is given by

$$f\left(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y} | \mathbf{p}_k, \mathbf{p}_l, H_k\right) = N_k! N_l! N_{\mathbf{y}}! \prod_{i=1}^{M} \frac{p_{k,i}^{x_{k,i}+y_i} p_{l,i}^{x_{l,i}}}{x_{k,i}! x_{l,i}! y_i!} \tag{1}$$

where (in the following notation k and l are exchangeable, and in this and subsequent chapters a boldface font is used to indicate vector quantities, while upper case is used for matrices)

k, $l \in$ {class 1, class 2}, and $k \neq l$;

H_k is the hypothesis defined as $\mathbf{p}_{\mathbf{y}} = \mathbf{p}_k$;

M is the number of discrete symbols;

$x_{k,i}$ is the number of occurrences of the i^{th} symbol in the training data for class k;

N_k $\left\{N_k = \sum_{i=1}^{M} x_{k,i}\right\}$ is the total number of training data for class k;

y_i is the number of occurrences of the i^{th} symbol in the test data;

$N_{\mathbf{y}}$ $\left\{N_{\mathbf{y}} = \sum_{i=1}^{M} y_i\right\}$ is the total number of test data;

$p_{k,i}$ $\left\{\sum_{i=1}^{M} p_{k,i} = 1\right\}$ is the probability of the i^{th} symbol for class k.

Note, a key assumption behind (1) is that the same underlying symbol distribution (i.e., $\mathbf{p}_k$ when class k is true) produces, independently, both training and test data. This is evident from $p_{k,i}^{x_{k,i}+y_i}$ where in the exponent training and test data are combined.

2.2.2 Combined Bayes Test

An important aspect of the CBT is that rather than assume $\mathbf{p}_k$ and $\mathbf{p}_l$ are simply unknown parameters to be estimated, and the resulting test a combined generalized likelihood ratio test (CGLRT) (the CGLRT represents the correctly-posed *generalized* likelihood ratio procedure which relies on ML probability estimates culled from both training and test data, see formula (41) in Appendix A), the approach here is to give them prior distributions. We assume nothing is known a priori about the probability vectors and so we use an "ignorance" prior. One version of prior ignorance is provided by the uniform Dirichlet given by, [34] (also see, [53]),

$$f\left(\mathbf{p}_k\right) = (M-1)!\mathcal{I}_{\left\{\sum_{i=1}^{M} p_{k,i}=1\right\}} \tag{2}$$

where $\mathcal{I}_{\{x\}}$ is the indicator function.

In the uniform Dirichlet distribution the symbol probabilities are *uniformly distributed on the unit hyperplane*, and it is obtained when all l_i (all l_i must be greater than zero) of the distribution,

$$f\left(p_{k,1},\ldots,p_{k,M};l_1,\ldots,l_M\right) = \frac{\Gamma\left(\sum_{i=1}^{M} l_i\right)}{\prod_{i=1}^{M}\Gamma\left(l_i\right)} p_{k,1}^{l_1-1} p_{k,2}^{l_2-1}\ldots p_{k,M}^{l_M-1}\mathcal{I}_{\left\{\sum_{i=1}^{M} p_{k,i}=1\right\}} \tag{3}$$

are set to unity, [5, 51]. Appendix B discusses the uniform Dirichlet further, and in particular, its marginal and conditional distributions are shown in formula (48).

Now, using the uniform Dirichlet the CBT appears as

$$\frac{f\left(\mathbf{x}_k,\mathbf{x}_l,\mathbf{y}|H_k\right)}{f\left(\mathbf{x}_k,\mathbf{x}_l,\mathbf{y}|H_l\right)} = \frac{\left(N_k+M-1\right)!\left(N_l+N_{\mathbf{y}}+M-1\right)!}{\left(N_k+N_{\mathbf{y}}+M-1\right)!\left(N_l+M-1\right)!}\prod_{i=1}^{M}\frac{\left(x_{k,i}+y_i\right)!\left(x_{l,i}\right)!}{\left(x_{k,i}\right)!\left(x_{l,i}+y_i\right)!} \underset{H_l}{\overset{H_k}{\gtrless}} \tau \tag{4}$$

where the decision threshold τ is equal to $P\left(H_l\right)/P\left(H_k\right)$ for minimizing the probability of error. Tests like the CBT shown above, which involve a ratio of posterior distributions, are in the statistics literature sometimes referred to as Bayes factors, [11, 38, 55].

The complete development of the CBT can be found in Appendix B, and it is included for completeness. However, the CBT can also be determined more straightforwardly (after correct substitution of model parameters, and a slight reworking of the result) from the Multinomial-Dirichlet distribution shown in [5]. In fact, the BDRA, which is developed later, is actually based on a conditional CBT equivalent to the Multinomial-Dirichlet. With this, it should also be noted that the CBT can easily be extended to classify more than two classes, and this is discussed in Chapter 3.

2.2.3 Probability of Error

Letting $z_k = f(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y}|H_k)$ (see formula (59) in Appendix B), the average probability of error for the CBT is defined as

$$P(e) = P(H_k) P(z_k \leq \tau z_l \mid H_k) + P(H_l) P(z_k > \tau z_l \mid H_l)_. \tag{5}$$

It is necessary to only show the first term of (5) as the second term is similar except for conditioning on H_l. Thus, ignoring $P(H_k)$, the first term of (5) is given by

$$P(z_k \leq \tau z_l \mid H_k) = \sum_{\mathbf{y}} \sum_{\mathbf{x}_k} \sum_{\mathbf{x}_l} \mathcal{I}_{\{z_k \leq \tau z_l\}} f(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y}|H_k) \tag{6}$$

where $f(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y}|H_k)$ is defined in formula (59) of Appendix B.

It is apparent from formulas (5) and (6) that computing the probability of error involves summations over all possible configurations of training and test data. Therefore, to reduce computational complexity this formula is worked further. Notice, a vector $\mathbf{y}$ representing $N_{\mathbf{y}}$ test samples contains $M_{\mathbf{y}}$ of M

possible discrete symbols (where $(M_{\mathbf{y}} \leq M)$). In other words, for a given sample of test data either all M discrete symbols are observed or some subset $M_{\mathbf{y}}$ are observed. Thus, the notation for training data is redefined as

$$\mathbf{x}_k = (\mathbf{x}_{kr}, \mathbf{x}_{kn})$$

with $\mathbf{x}_{kr}$ referring to those training data which are "represented" by the same $M_{\mathbf{y}}$ discrete symbols as the test data, and $\mathbf{x}_{kn}$ to those which are not. Because of this, it is important to note that the indicator function, $\mathcal{I}_{\{z_k \leq \tau z_l\}}$, of (6) depends only on $\mathbf{y}$, $\mathbf{x}_{kr}$, and $\mathbf{x}_{lr}$. Therefore, summations over $\mathbf{x}_{kn}$ and $\mathbf{x}_{ln}$ (these summations are not required if $M_{\mathbf{y}} = M$) can precede first in formula (6) so that it becomes

$$P\left(z_k \leq \tau z_l \mid H_k\right) = \sum_{\mathbf{y}} \sum_{\mathbf{x}_{kr}} \sum_{\mathbf{x}_{lr}} \mathcal{I}_{\{z_k \leq \tau z_l\}} \sum_{\mathbf{x}_{kn}} \sum_{\mathbf{x}_{ln}} f\left(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y} | H_k\right) \qquad (7)$$

where, using the formulas (57) and (58) of Appendix B, the summations over $\mathbf{x}_{kn}$ and $\mathbf{x}_{ln}$ produce

$$\begin{aligned}
&\sum_{\mathbf{x}_{kn}} \sum_{\mathbf{x}_{ln}} f\left(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y} | H_k\right) \\
&\quad = \sum_{\mathbf{x}_{kn}} \sum_{\mathbf{x}_{ln}} \frac{\left[(M-1)!\right]^2 N_k! N_l! N_{\mathbf{y}}!}{(N_k + N_{\mathbf{y}} + M - 1)!(N_l + M - 1)!} \prod_{i \in ir} \prod_{j=1}^{y_i} \frac{(x_{k,i} + j)}{y_i!} \\
&\quad\quad = \frac{\left[(M-1)!\right]^2 N_k! N_l! N_{\mathbf{y}}!}{(N_k + N_{\mathbf{y}} + M - 1)!(N_l + M - 1)!} \prod_{i \in ir} \prod_{j=1}^{y_i} \frac{(x_{k,i} + j)}{y_i!} \\
&\times \binom{N_k - \sum \mathbf{x}_{kr} + M - M_{\mathbf{y}} - 1}{N_k - \sum \mathbf{x}_{kr}} \binom{N_l - \sum \mathbf{x}_{lr} + M - M_{\mathbf{y}} - 1}{N_l - \sum \mathbf{x}_{lr}}. \qquad (8)
\end{aligned}$$

With respect to notation, $\sum \mathbf{x}_{kr}$ above means the sum of all training data under k that are represented by the same $M_{\mathbf{y}}$ discrete symbols as test data. Also, the term "ir" in formula (8) refers to that subset of symbols $\{1, 2, ..., M\}$ contained in the test data. With this, observe that $\begin{pmatrix} N_k - \sum \mathbf{x}_{kr} + M - M_{\mathbf{y}} - 1 \\ N_k - \sum \mathbf{x}_{kr} \end{pmatrix}$ means the number of ways $N_k - \sum \mathbf{x}_{kr}$ training data can be arranged amongst $M - M_{\mathbf{y}}$ discrete symbols. Further, it is important to note that when $M = M_{\mathbf{y}}$ both factors of the form, $\begin{pmatrix} N_k - \sum \mathbf{x}_{kr} + M - M_{\mathbf{y}} - 1 \\ N_k - \sum \mathbf{x}_{kr} \end{pmatrix}$, are dropped from formula (8).

In addition to that shown above, the computational complexity associated with computing $P(e)$ in the formula of (5) can be further reduced by exploiting redundancy in the test data. This has the effect of reducing the required number of terms to be summed over. The method to do this is based on the fact that for a given number $N_{\mathbf{y}}$ of test observations, there are a finite number of unique $\mathbf{y}$'s (each vector taking on a varying number, $M_{\mathbf{y}}$, of discrete symbols) that are possible. All of the remaining $\mathbf{y}$'s have elements that are redundant orderings of the original set, and these orderings occur in a countable number of ways. For example, an important simplification used throughout the results obtained in this work is when $N_{\mathbf{y}} = 1$. In this case, formula (8) becomes

$$\sum_{\mathbf{x}_{kn}} \sum_{\mathbf{x}_{ln}} f(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y} | H_k) = \frac{[(M-1)!]^2 N_k! N_l!}{(N_k + M)!(N_l + M - 1)!}(x_{k,ir} + 1)$$

$$\times \begin{pmatrix} N_k - \sum \mathbf{x}_{kr} + M - 2 \\ N_k - \sum \mathbf{x}_{kr} \end{pmatrix} \begin{pmatrix} N_l - \sum \mathbf{x}_{lr} + M - 2 \\ N_l - \sum \mathbf{x}_{lr} \end{pmatrix}. \tag{9}$$

Note, because of symmetry in the Dirichlet distribution formula (9) is equal for all "ir" in $\{1, 2, ..., M\}$. This implies that when $N_{\mathbf{y}} = 1$ the summation over $\mathbf{y}$ in formula (7) involves a sum of the same M terms. Also, the summations over $\mathbf{x}_{kr}$ and $\mathbf{x}_{lr}$ in formula (7) are then defined as, respectively, $\sum_{x_{k,i}=1}^{N_k}$ and $\sum_{x_{l,i}=1}^{N_l}$.

2.3 Results

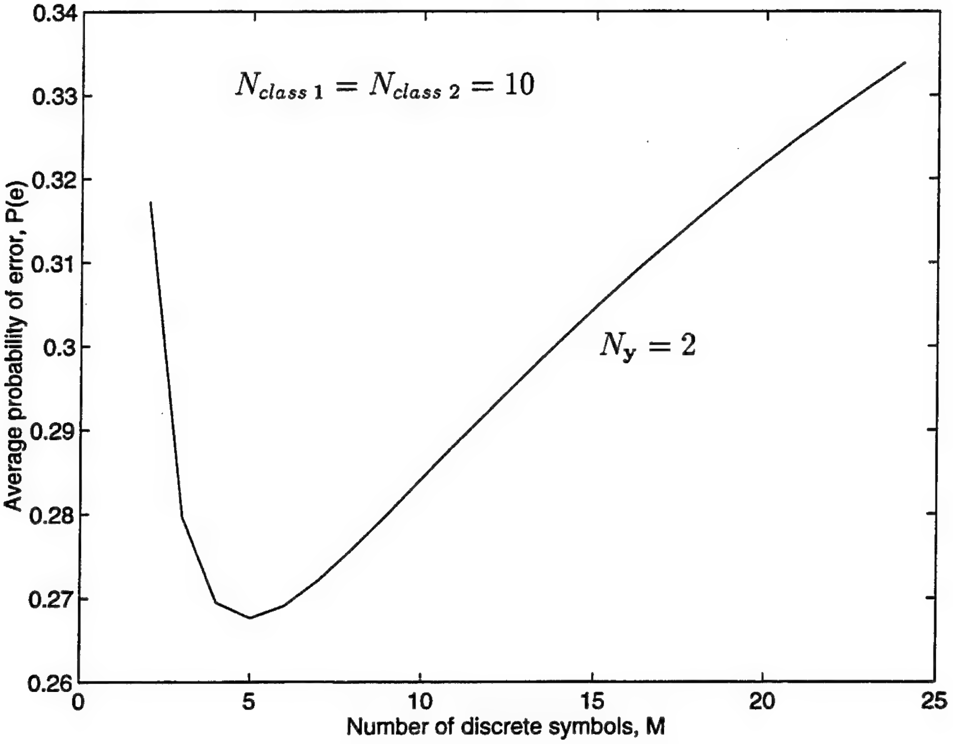
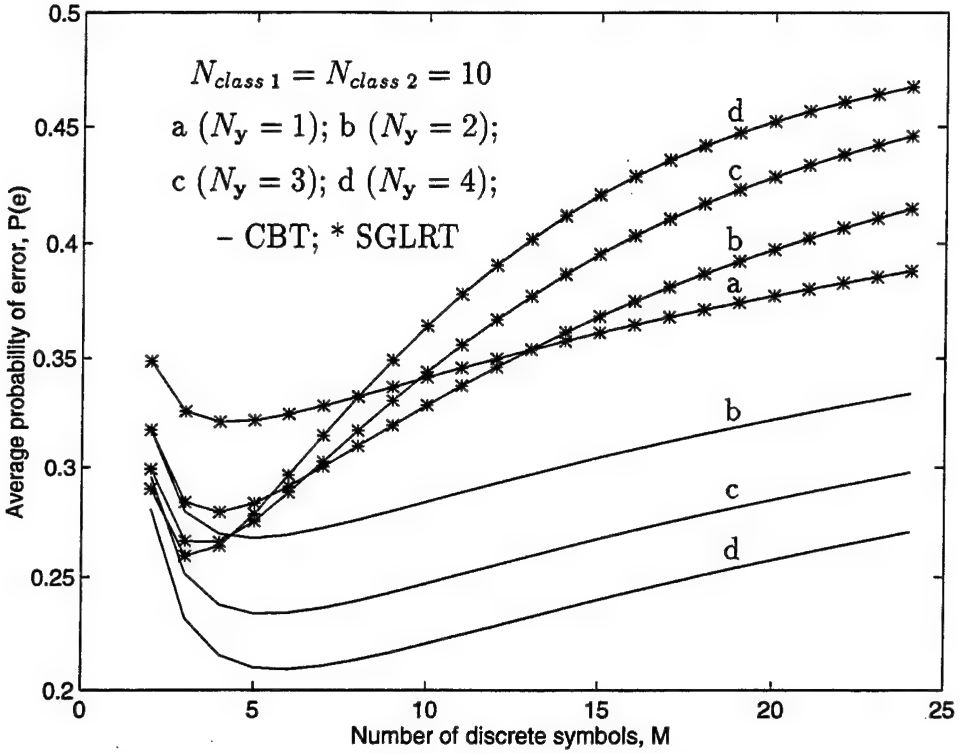


Figure 3: $P(e)$ for various numbers of discrete symbols M.

Figure 3 is a typical plot of the results presented here, and unless otherwise indicated, in all figures contained in this dissertation the threshold $\tau = 1$, meaning that $P(H_k)$ and $P(H_l)$ are both 0.5. Also, performance results for the CBT using simulated data are shown in Appendix A. In Figure 3, appears the average probability of error for the CBT, with varying M, using the formula shown in (5). In this example, $N_{class\,1} = N_{class\,2} = 10$, and $N_{\mathbf{y}} = 2$. Notice in Figure 3 that $P(e)$ starts out decreasing with increasing M and is minimum at a point called M^* (see [34]), and in this case $M^* = 5$. Further, it can be seen that for M greater than M^* $P(e)$ steadily increases. Observe, $P(e)$ is upper bounded by 0.5 at $M = 1$ and at $M = \infty$ where it is determined by the priors, $P(H_k)$ and $P(H_l)$, that is, at these points the training data cannot be relied on, [23]). In [61], the dependence of $P(e)$ on M has been attributed to incomparable models. That is, each value of M represents a different prior for the specified numbers of training and test data, and as such performance cannot be compared. However, the interest here is in problems where no prior knowledge exists neither about the discrete symbol probabilities, nor about the "correct" quantization complexity (M). Thus, within this context, the result shown above implies that on average, and with ten samples of training data for each class and two test observations, best classification performance (i.e., minimum $P(e)$) occurs when five discrete symbols are used. In other words, if more than five symbols are used, then the probability of a test symbol being unrepresented in the training data increases, and this causes more classification errors. On the other hand, if not enough discrete symbols are used, then the true probabilistic structure of each class may not be adequately represented, and this also increases $P(e)$. When faced with uncertainty, such information is useful as a guide in selecting the most favorable quantization complexity for particular training and test data sizes.



(The curves for $N_{\mathbf{y}} = 1$ are identical for the two schemes.)

Figure 4: $P(e)$ for the CBT and the SGLRT.

Figure 4 illustrates the performance gains of including test data to infer symbol probabilities by comparing $P(e)$ for the CBT to that of an uncombined generalized likelihood ratio type test (referred to as the standard GLRT, or SGLRT).

The SGLRT, being uncombined, does not incorporate test data to infer symbol probabilities, and it is given by (performance results for the SGLRT using simulated data appear in Appendix A)

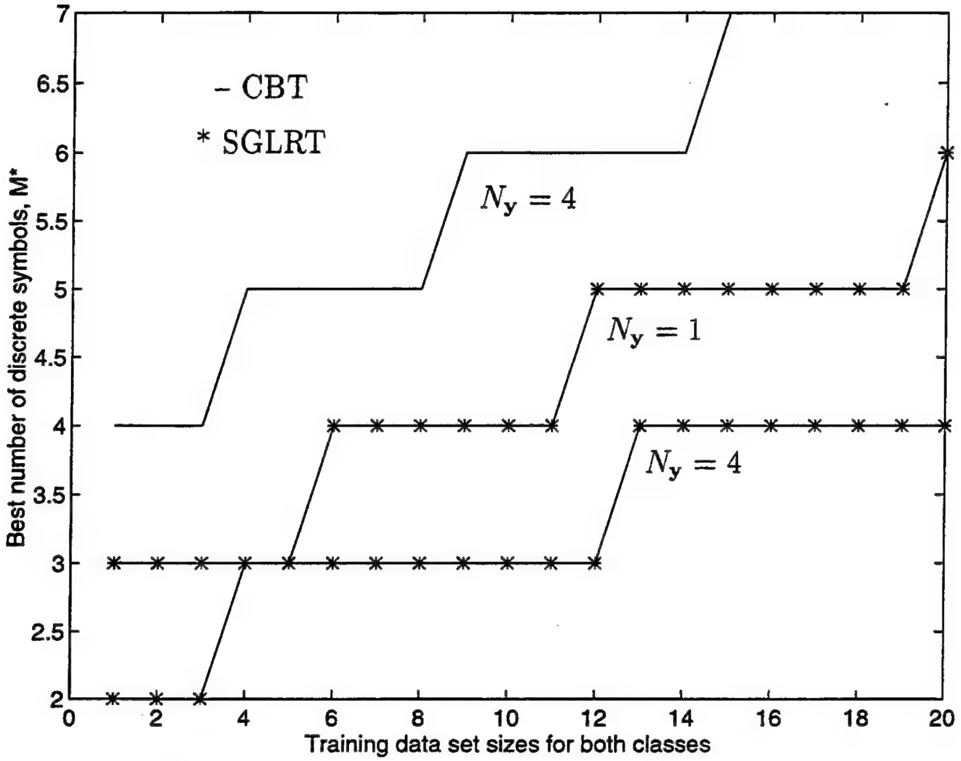
$$\frac{f\left(\mathbf{y}|\hat{\mathbf{p}}_k, H_k\right)}{f\left(\mathbf{y}|\hat{\mathbf{p}}_l, H_l\right)} = \prod_{i=1}^{M} \frac{\hat{p}_{k,i}^{y_i}}{\hat{p}_{l,i}^{y_i}} \underset{H_l}{\overset{H_k}{\gtrless}} \tau \tag{10}$$

where, using maximum likelihood estimates,

$$\hat{p}_{k,i} = \frac{x_{k,i}}{N_k} \text{ and } \hat{p}_{l,i} = \frac{x_{l,i}}{N_l}. \tag{11}$$

In the situation of Figure 4 the training data of each class are fixed to ten outcomes, and four different test data sizes are used. It can be seen in this figure that when $N_{\mathbf{y}} = 1$ both tests perform identically (this was previously pointed by Nádas, [52]), and $M^* = 4$. However, when $N_{\mathbf{y}}$ is increased the situation changes quite rapidly. The CBT consistently shows an overall relative decrease in $P(e)$ for a given M, while the SGLRT performs worse except for small values of M. With this, when $N_{\mathbf{y}}$ is increased to four, for the CBT M^* increases to six, whereas for the SGLRT it decreases to three.

A few additional comments are included to supplement these results. Notice, because the CBT uses both training and test data to infer symbol probabilities the additional information provided by the test data causes a relative decrease in $P(e)$, and an accompanying increase in M^*. Also, intuitively, the probability a test datum takes on a discrete symbol unrepresented by training data increases with $N_\mathbf{y}$ and M, which accounts for the SGLRT's poor performance as both numerator and denominator of the test shown in (10) will then have an increasing probability of being zero valued. Further, in Appendix D it is shown that the minimum $P(e)$ obtainable with a Dirichlet distribution on the symbol probabilities, and when $N_\mathbf{y} = 1$, is equal to 0.25. But, observe for the CBT in Figure 4 that additional test observations allow $P(e)$ to be reduced below this value.



(Note, both tests perform the same when $N_\mathbf{y} = 1$.)

Figure 5: M^* for various training data set sizes.

Figure 5 is used to illustrate the relationship between best discrete symbol quantity and training data size. Shown for both the CBT and the SGLRT is a plot of M^* versus training data sizes of up to twenty outcomes ($N_{class\ 1} = N_{class\ 2}$).

Two different numbers of test observations appear for both tests, and they are $N_{\mathbf{y}} = 1$ and $N_{\mathbf{y}} = 4$. In this figure, it is apparent for both tests that when the number of training data is increased a larger number of discrete symbols are required for best classification performance (again, both tests perform the same when $N_{\mathbf{y}} = 1$). But, for the CBT the rate of increase in the required number of symbols is faster when $N_{\mathbf{y}} = 4$ than it is when $N_{\mathbf{y}} = 1$, while, for the SGLRT the opposite is true (recall Figure 4 where performance of the SGLRT diminishes with increasing $N_{\mathbf{y}}$).

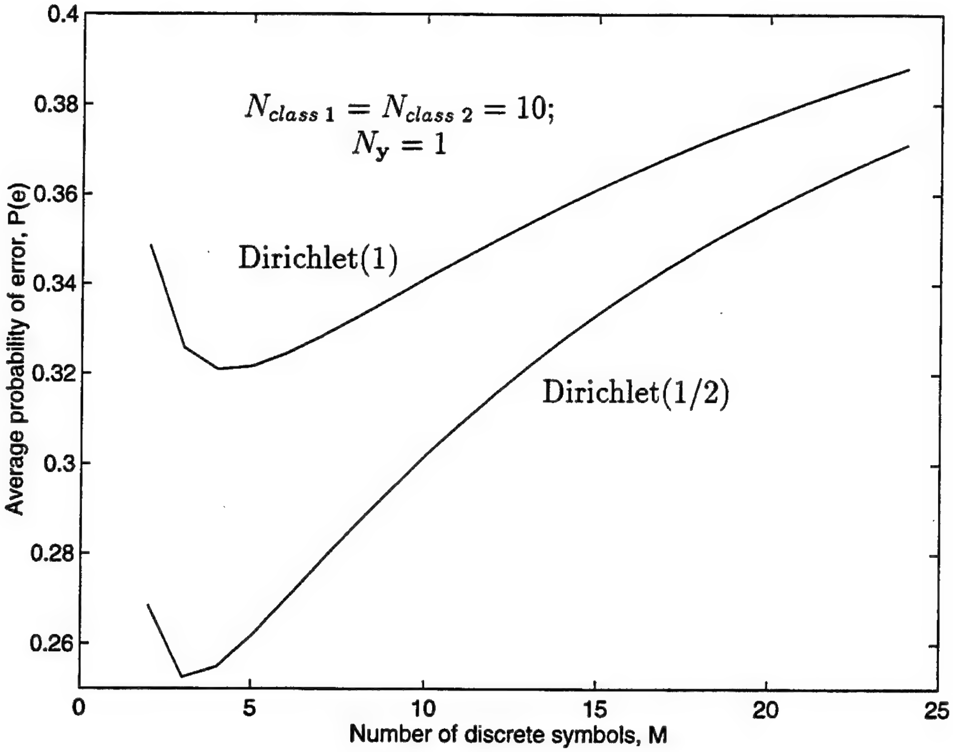


Figure 6: Comparison of $P(e)$ using different Dirichlet priors.

Before proceeding to the next section, notice in previous work, [42], it was found that in relation to universal encoding a better noninformative prior to use, given unknown true statistics, is the Dirichlet with all l_i of formula (3) set to one half. In the literature, this distribution is also known as Jeffreys prior for the multinomial (see, [10]). Observe, in Figure 6 appears curves of the average probability of error for the two cases of a uniform Dirichlet (i.e., Dirichlet(1), which is replotted from Figure 4), and a Dirichlet with its parameters set to one half (i.e., Dirichlet(1/2)). With this, both curves in Figure 6 are based on ten samples of training data for each class and one test observation. Notice, it can be seen in this figure that results based on the Dirichlet(1/2) are better than those based the uniform Dirichlet. In fact, the value of M^* when using the Dirichlet(1/2) is one less than it is when using the uniform Dirichlet, and this indicates that less feature information is required for best performance. However, although the Dirichlet(1/2) shows better overall average performance, it does not treat each symbol probability equally (actually, the Dirichlet(1/2) puts more weight on probabilities near zero and one, see, [11]) so that the uniform Dirichlet is used here to better represent complete ignorance about the underlying symbol probabilities of each class. Also, use of the uniform Dirichlet for this application is consistent with previous work (e.g., [34, 61]), and as will be seen in Chapter 3 it is more effective than the Dirichlet(1/2) at data reduction.

2.4 Testing the Statistical Similarity of Discrete Data

In addition to using the CBT of (4) for classifying an unknown test data vector, it can also be used to test if two samples of discrete data are produced by the same multinomial distribution. Analogously, this is the same as testing the statistical similarity of two histograms. Based on (1), the joint distributions for the number of occurrences of all symbols for both data sets given they are produced by the same probability vector $\mathbf{p}_k$ (i.e., $H_1 : \mathbf{p}_k = \mathbf{p}_l$), and given they are produced by independent probability vectors $\mathbf{p}_k$ and $\mathbf{p}_l$ (i.e., $H_0 : \mathbf{p}_k \neq \mathbf{p}_l$) are given by, respectively,

$$f\left(\mathbf{x}_k, \mathbf{x}_l | \mathbf{p}_k, \mathbf{p}_l, H_1\right) = N_k! N_l! \prod_{i=1}^{M} \frac{p_{k,i}^{x_{k,i}+x_{l,i}}}{x_{k,i}! x_{l,i}!} \tag{12}$$

and

$$f\left(\mathbf{x}_k, \mathbf{x}_l | \mathbf{p}_k, \mathbf{p}_l, H_0\right) = N_k! N_l! \prod_{i=1}^{M} \frac{p_{k,i}^{x_{k,i}} p_{l,i}^{x_{l,i}}}{x_{k,i}! x_{l,i}!} \tag{13}$$

where, using notation similar to (1)

k, $l \in$ {sample 1, sample 2}, and $k \neq l$;

M is the number of discrete symbols (or histogram bins);

$x_{k,i}$ is the number of occurrences of the i^{th} symbol for sample k;

N_k $\left\{N_k = \sum_{i=1}^{M} x_{k,i}\right\}$ is the total number of occurrences of the M symbols for sample k;

$p_{k,i}$ $\left\{\sum_{i=1}^{M} p_{k,i} = 1\right\}$ is the probability of the i^{th} symbol.

A CBT for this application can be developed using the same procedure shown in Appendix B. However, it can also be obtained directly from formula (4). This is accomplished by first eliminating any training data under class l, followed by renaming the y_i as $x_{l,i}$, and then redefining the hypotheses H_1 and H_0 by those shown above. In either case, the test for this situation appears as

$$\frac{f\left(\mathbf{x}_k, \mathbf{x}_l | H_1\right)}{f\left(\mathbf{x}_k, \mathbf{x}_l | H_0\right)} = \frac{(N_k + M - 1)!\,(N_l + M - 1)!}{(M-1)!\,(N_k + N_l + M - 1)!} \prod_{i=1}^{M} \frac{(x_{k,i} + x_{l,i})!}{(x_{k,i})!\,(x_{l,i})!} \underset{H_0}{\overset{H_1}{\gtrless}} \tau. \tag{14}$$

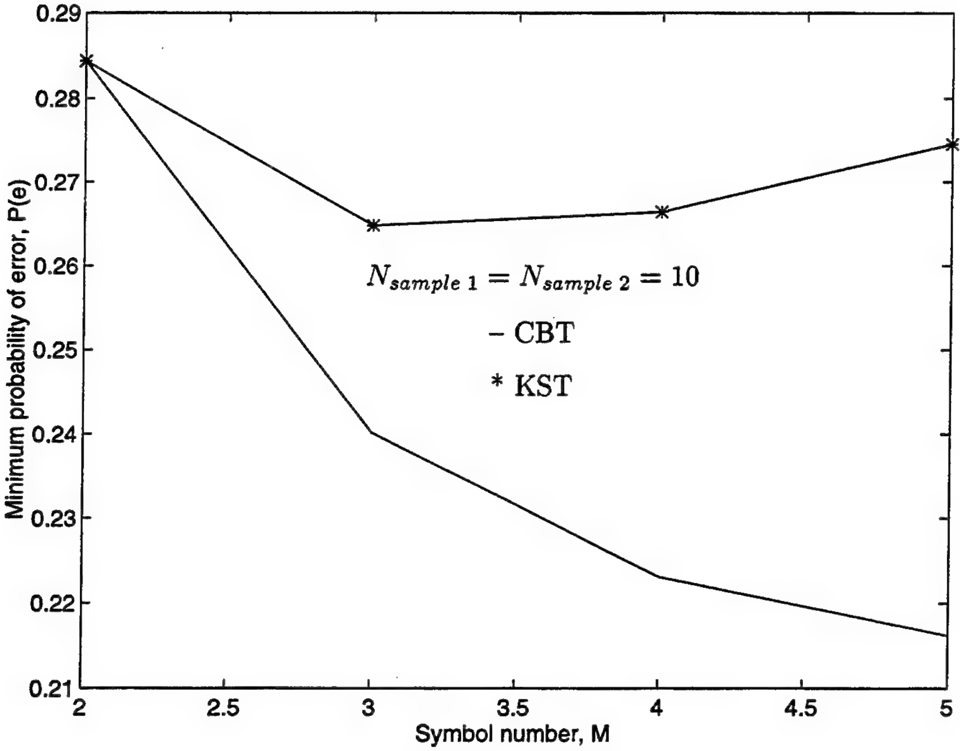


Figure 7: The minimum analytical $P(e)$ for the CBT and the KST.

In Figure 7, the CBT of formula (14) is compared to the Kolmogorov-Smirnov Test (KST), [15]. Using the notation of (12) through (14), the KST is defined as

$$\frac{1}{\sup_{1\leq w\leq M}\left|F_{N_k}(w)-F_{N_l}(w)\right|\left(\frac{N_kN_l}{N_k+N_l}\right)^{1/2}} \underset{H_0}{\overset{H_1}{\gtrless}} \tau \tag{15}$$

where $F_{N_k}(w)$ represents the cumulative distribution function.

It is also noted that a Chi-Square Goodness of Fit Test, [33], can also be used in this application. However, like the CGLRT it is in general only asymptotically optimal, and it must be modified if a discrete symbol is unobserved in the data.

The advantage of using the test of (14) in place of the test of (15) is clearly demonstrated in Figure 7. In this figure, the minimum average probability of error in testing if two sets of discrete data belong to the same multinomial distribution is plotted for both tests versus the number of symbols (M). Notice, results are shown for symbol quantities of from two to five symbols, and each population contains ten samples of data. Also, by minimum average probability of error it is meant that for each discrete symbol quantity the least $P(e)$ over all possible thresholds was chosen and plotted in Figure 7. Given this, it can be seen that both tests start out with the same $P(e)$ when $M = 2$, but by $M = 5$, $P(e)$ is steadily decreasing for the CBT, while it is increasing for the KST. With this, observe in Figure 7 that M^* occurs at three discrete symbols for the KST, and for the CBT M^* appears to be much larger. In this case, due to computational complexity M^* was not determined theoretically for the CBT, however, empirically M^* was estimated to be approximately 12. In general, it can be seen that the superior performance of the CBT allows for more precise threshold settings when testing at a specified probability of false alarm. But, similar to the GLRT for classification, performance of the KST will asymptotically approach that of the CBT as the sample sizes become large (see Appendix A).

2.5 Summary

In this chapter, it has been demonstrated that given only the training and test data set sizes, and without any a priori knowledge of the underlying symbol probabilities of each class, it is possible to determine the discrete symbol quantity which minimizes the average probability of error. Specifically, the number of discrete symbols achieving this minimum point was called M^*. Further, it was shown that M^* increases with the training data size for both the CBT and the SGLRT. However, rates of increase in M^* are higher with the number of test data, and overall $P(e)$ lower, only for the CBT. Additionally, the CBT was shown to achieve a lower minimum average probability of error than the KST (except when $M = 2$ where they are equal) for testing if two discrete data sets are statistically similar.

Chapter 3

The Bayesian Data Reduction Algorithm

3.1 Introduction

The focus of this chapter is on developing the Bayesian Data Reduction Algorithm (BDRA), which is based on the CBT of Chapter 2. The BDRA uses a "greedy" approach for reducing features from the training data of each class, and it relies on the conditional probability of error for the CBT (formula (5) of Chapter 2 conditioned on the training data) as a metric for making data reducing decisions. Performance of the algorithm is compared to a neural network at classifying discrete feature vectors which contain binary and ternary valued features. In this comparison, it is shown that the BDRA is superior to the neural network at improving overall classification performance by reducing, or eliminating, irrelevant feature information. However, for a fixed amount of training data, the performance of both schemes degrades as the quantization complexity of each feature is increased from binary to ternary values.

3.2 Development of the BDRA

A fundamental component of the BDRA is the conditional probability of error formula for the CBT. But, before developing this formula, and its associated decision rule, for convenience the notation used here is itemized first (see formula (1) of Chapter 2):

- C is the total number of classes with $k \in \{1, \ldots, C\}$.
- M is the number of discrete symbols.
- H_k is the hypothesis defined as $\mathbf{p_y} = \mathbf{p}_k$;
- $X \equiv (\mathbf{x}_1, \ldots, \mathbf{x}_C)$ is the collection of training data from all C classes.
- $x_{k,i}$ is the number of occurrences of the i^{th} symbol in the training data for class k.
- N_k $\left\{N_k = \sum_{i=1}^{M} x_{k,i}\right\}$ is the total number of training data for class k.
- y_i is the number of occurrences of the i^{th} symbol in the test data.
- $N_\mathbf{y}$ $\left\{N_\mathbf{y} = \sum_{i=1}^{M} y_i\right\}$ is the total number of test data.
- $p_{k,i}$ $\left\{\sum_{i=1}^{M} p_{k,i} = 1\right\}$ is the probability of the i^{th} symbol for class k.
- $\mathcal{I}_{\{x\}}$ is the indicator function.

The conditional probability of error for the CBT depends on a decision rule that decides if an unknown test vector, $\mathbf{y}$, belongs to a class k given knowledge of the training data. Thus, the distribution $f(\mathbf{y}|X, H_k)$ must be found. However, based on the assumption that the training data of each class is independent of the other training data sets (e.g., x_l is independent of H_k), this distribution is equivalent to $f(\mathbf{y}|\mathbf{x}_k, H_k)$. Therefore, with equiprobable classes, the decision rule for a conditional CBT is then given by (a similar rule without specifying distributions was shown in [46]),

$$\max_{1 \le k \le C} \left[f(\mathbf{y}|\mathbf{x}_k, H_k) \right] \tag{16}$$

where ties are broken arbitrarily.

Now, it is straightforward to see that with independent training data sets for each class the distribution $f(\mathbf{y}|\mathbf{x}_k, H_k)$ can be obtained from the ratio of $f(\mathbf{y}, \mathbf{x}_k|H_k)$ and $f(\mathbf{x}_k)$, where the later distributions appear respectively in Appendix B as formulas (57) and (58) (defined for H_k instead of H_l). Thus, the conditional distribution of formula (16) can be written as

$$f(\mathbf{y}|\mathbf{x}_k, H_k) = \frac{(N_k + M - 1)!\,(N_{\mathbf{y}})!}{(N_k + N_{\mathbf{y}} + M - 1)!} \prod_{i=1}^{M} \frac{(x_{k,i} + y_i)!}{(x_{k,i})!\,(y_i)!}. \tag{17}$$

Note, for completeness, formula (17) is also developed from probabilistic considerations in Appendix C. Further, a consequence of independent training data sets amongst the classes is that formulas (16) and (17) represent the extension of the CBT of formula (4) in Chapter 2 to C classes.

Given the results above, and letting $z_k = f(\mathbf{y}|\mathbf{x}_k, H_k)$, the associated conditional probability of error formula for the test of (16) is given by

$$P(e \mid X) = \sum_{k=1}^{C} \sum_{\mathbf{y}} \sum_{\mathbf{x}_k} \sum_{\mathbf{x}_l} P(H_k)\, \mathcal{I}_{\{z_k \leq z_l, \text{ for all } k \neq l\}} f(\mathbf{y}|\mathbf{x}_k, H_k) \tag{18}$$

The results in this chapter are based on one observation of test data, therefore, with $N_{\mathbf{y}} = 1$ formula (17) becomes

$$f(y_i = 1|\mathbf{x}_k, H_k) = \frac{x_{k,i} + 1}{N_k + M}, \tag{19}$$

where $i \in \{1, \ldots, M\}$.

Notice, results appear only for the case of $N_{\mathbf{y}} = 1$ because this is sufficient to illustrate all key aspects of the BDRA's performance. However, for classification situations in which $N_{\mathbf{y}} > 1$ it is expected that the BDRA will be applied with the appropriate number of test observations. Then, and although this has not been fully studied here, consistent with Figure 4 of Chapter 2 the additional test observations should cause a relatively lower $P(e \mid X)$ (and a larger final quantization complexity) as compared to when $N_{\mathbf{y}} = 1$. Otherwise, for any $N_{\mathbf{y}}$ the general relative performance characteristics of the BDRA should remain consistent with those shown below.

Now, with $P(e \mid X)$ defined in formula (18), the following iterative steps are used in implementing the BDRA.

1. Using the initial training data with quantization complexity M (e.g., in the case of all binary valued features $M = 2^{N_f}$, where N_f is the number of features), formula (18) is used to compute $P(e \mid X; M)$.

2. Beginning with the first feature (selection is arbitrary), reduce this feature for each class by summing (i.e., merging) the numbers of occurrences of those quantized symbols that correspond to joining adjacent discrete levels of that feature (e.g., with binary features, for all classes merge those quantized symbols containing a binary zero for that feature with those containing a binary one).

3. Use the newly merged training data (it is referred to as X') and the new quantization complexity (e.g., $M' = 2^{N_f - 1}$ in the binary feature case), and use formula (18) to compute $P\left(e \mid X'; M'\right)$.

4. Repeat items two and three for all N_f features.

5. From item four select the minimum of all computed $P\left(e \mid X'; M'\right)$ (in the event of a tie use an arbitrary selection), and choose this as the new training data configuration for each class (this corresponds to permanently reducing, or removing, the associated feature).

6. Repeat items two through five until the probability of error does not decrease any further, or until $M' = 2$, at which point the final quantization complexity has been found.

A few additional notes about the BDRA are necessary. First, the BDRA is "greedy" in that it chooses a best training data configuration at each iteration (see step five above) in the process of determining a best quantization complexity.

In fact, the six steps of the BDRA shown above are analogous to what is known as a backward sequential search algorithm, [37, 40]. Also, a slightly better approach than the six steps shown above is to do a global search over all possible merges and corresponding training data configurations. However, a simulation study involving hundreds of independent trials revealed that only about three percent of the time did the "greedy" approach shown here produce results different than a global approach. Additionally, the overall average probability of error for the two approaches differed by only an insignificant amount.

With this, it should also be noted that data reduction in the BDRA implies a change in Dirichlet prior (see formula (2) of Chapter 2) with each merging of training data (i.e., due to a reduction in M). But, it is argued here that changing the Dirichlet prior by the BDRA is justified because it essentially removes irrelevant feature information with each merge. In other words, the BDRA looks for that quantization complexity, M, which makes the most sense based on the training data.

Before presenting performance results of the BDRA observe that as a check on a Bayesian approach to this problem other data reduction algorithms were also developed based on the SGLRT of formula (10), Chapter 2, and the CGLRT of formula (41) in Appendix A. Development of these algorithms consisted of substituting the appropriate distributional formula in the z_k of formula (18). That is, in the case of the SGLRT $z_k = f(\mathbf{y}|\hat{\mathbf{p}}_k, H_k)$, and in the case of the CGLRT $z_k = f(\mathbf{x}_k, \mathbf{y}|\hat{\mathbf{p}}_k, H_k) / f(\mathbf{x}_k|\hat{\mathbf{p}}_k, H_k)$. But, in implementation it was found that neither of these GLR methods would work as well as the CBT in data reduction. In fact, in all of the cases examined (i.e., situations of the type shown in the results below) an SGLRT based data reduction method could not eliminate a single irrelevant feature. On the other hand, for the same training data sets the BDRA was able to effectively reduce the data as shown below.

3.3 Results

Performance results of the BDRA appear in the figures found on the following pages. In all cases, one test observation is used ($N_{\mathbf{y}} = 1$), and there are two classes (i.e., $C = 2$). Note, the results presented in each figure are averaged over one hundred independent trials of randomly generating symbol probabilities and associated training data. With this, the training data sets of each class contain either six binary valued features or six ternary valued features so that the initial unreduced quantization complexity, M, either equals 64 or 729, respectively.

The following items define the notation used for the error probabilities shown in the figures below:

Unmerged (Training Data) The probability of error computed using formula (18) and the initial training data configuration of each class before data reduction.

Merged (Training Data) The probability of error computed using formula (18) and the final reduced training data configuration for each class (i.e., after applying the BDRA).

Optimal The probability of error computed when the true underlying symbol probabilities are known.

Unmerged (True) The probability of error computed using formula (18) with the z_k based on the initial unmerged training data (and formula (19)), and $f(\mathbf{y}|\mathbf{x}_k, H_k)$ replaced by the true symbol probabilities.

Merged (True) The probability of error computed using formula (18) with the z_k based on the merged training data (and formula (19) after applying the BDRA), and $f(\mathbf{y}|\mathbf{x}_k, H_k)$ replaced by the true symbol probabilities.

Neural Network The probability of error computed using a trained neural network as a decision rule, and the true symbol probabilities.

3.3.1 Performance at Reducing Binary Valued Irrelevant Features

Figure 8 below shows error probabilities of the BDRA as a function of the number of relevant features for each class. The term "relevant" means that those features are distributed uniquely for each class with the remaining features, out of the total of six, being distributed the same amongst the classes. Note, that the results in Figure 8 are based on randomly generating twenty five samples of training data for each class, where each sample is a vector containing six

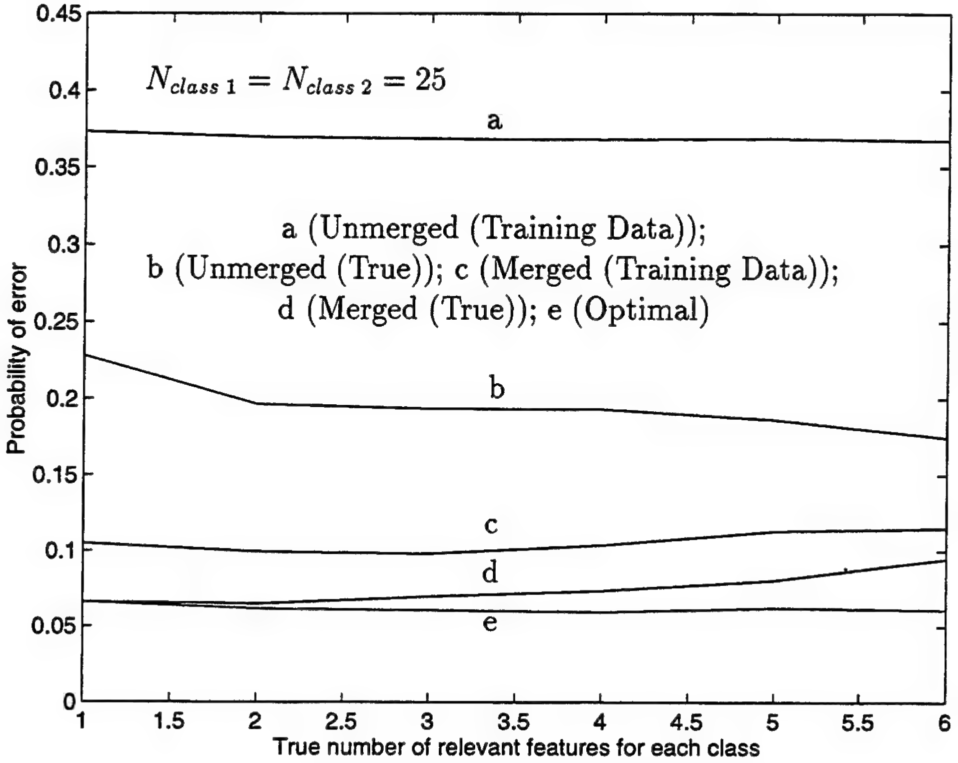


Figure 8: Performance of the BDRA with six binary valued features, and twenty five samples of training data for each class.

binary valued features. Observe, that twenty five samples of training data is a relatively small number of samples to estimate the probabilities of sixty four discrete symbols, and this is intended to make correct classification more difficult.

It can be seen in Figure 8 that based on formula (18), and for all numbers of relevant features, the BDRA starts out with a probability of error of more than 0.35 (Unmerged (Training Data)), and then reduces this to around 0.1 (Merged (Training Data)). But, in terms of what this implies for correct classification performance, notice that based on true statistics the algorithm is able to reduce the probability of error from about 0.2 (Unmerged (True)) to near optimal (Merged

(True)). In fact, with one relevant feature for each class it obtains the optimal probability of error. However, there also is somewhat of a relative loss in performance for the BDRA as the number of relevant features increases, and this issue is addressed later in Figure 12.

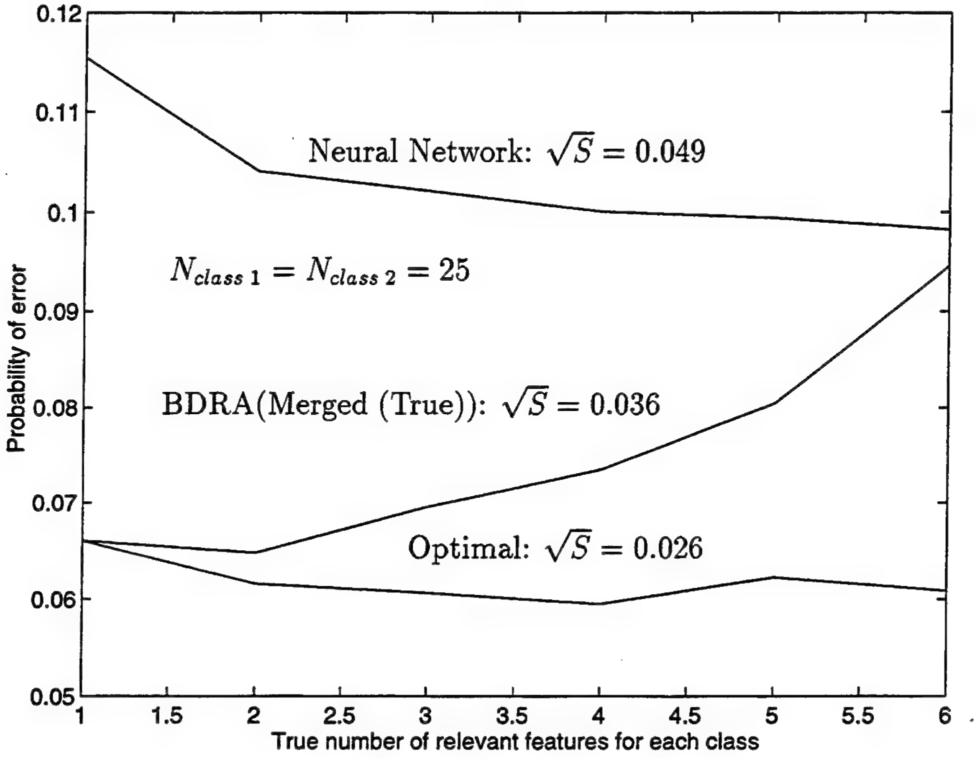


Figure 9: Performance comparison of the BDRA to a neural network with binary valued features, and twenty five samples of training data for each class.

Performance of the BDRA is compared to a neural network in Figure 9. The situation in this figure is the same as that of Figure 8, and it can be seen that the Optimal and Merged (True) results from Figure 8 are repeated here. Notice, also in Figure 9 are classification results for a neural network, and the term $\sqrt{S}$ represents the sample standard deviation, [12], for the probability of error averaged over all numbers of relevant features. Clearly, it can be seen that in all cases the BDRA is superior to the neural network by achieving a lower probability of error, and a smaller sample standard deviation. But, the BDRA appears to deliver best performance when the number of relevant features is minimum, whereas the opposite is true for the neural network.

In generating results for the neural network it was trained and tested using the Neural Network Toolbox of Matlab, [16]. It is a feed-forward network (whose neuron model is the log-sigmoid transfer function), which was trained using back-propagation, momentum, and an adaptive learning rate. The network was specified to contain three layers including two successive hidden layers consisting of sixteen and eight nodes, and an output layer of one node. The input consisted of six nodes corresponding to the the six discrete features, and initialization of network weights was random.

The following items describe the relevant neural network parameter settings required by the Matlab software:

- Maximum number of epochs to train (1000).
- Sum-squared error goal (0.02).
- Learning rate (0.01).
- Learning rate increase when adapting (1.05).
- Learning rate decrease when adapting (0.7).

- Momentum constant (0.9).

- Maximum error ratio (1.04).

Additionally, and this is relevant to most of the results in this chapter, with all figures the optimal probabilities of error were constrained to be less than or equal to 0.1 (in some cases, and this is pointed out where applicable, the constraint is also defined to be greater than or equal to 0.5 as well). To achieve this constraint it was necessary to use Gaussian mixture densities for generating the underlying true symbol probabilities. In doing so, two equiprobable Gaussian mixtures were used for relevant features, and three equiprobable mixtures were used for irrelevant ones (the dimension of a particular Gaussian pdf was equivalent to the appropriate number of features). Thus, the probability of observing a binary one for a feature was equivalent to the probability of observing a positive value for the associated element of the corresponding Gaussian mixture. Using this model, controlling the probability of error to meet the specified constraint was done by adjusting the spread of the means. The reason Gaussian mixtures were chosen instead of the Dirichlet distribution, or a similar uniform type distribution, was that the probability of error using the Dirichlet converges to 0.25 with symbol quantity, M, while its variance approaches zero (see Appendix D). Therefore, constraining the probability of error to small values using the Dirichlet was computationally impractical.

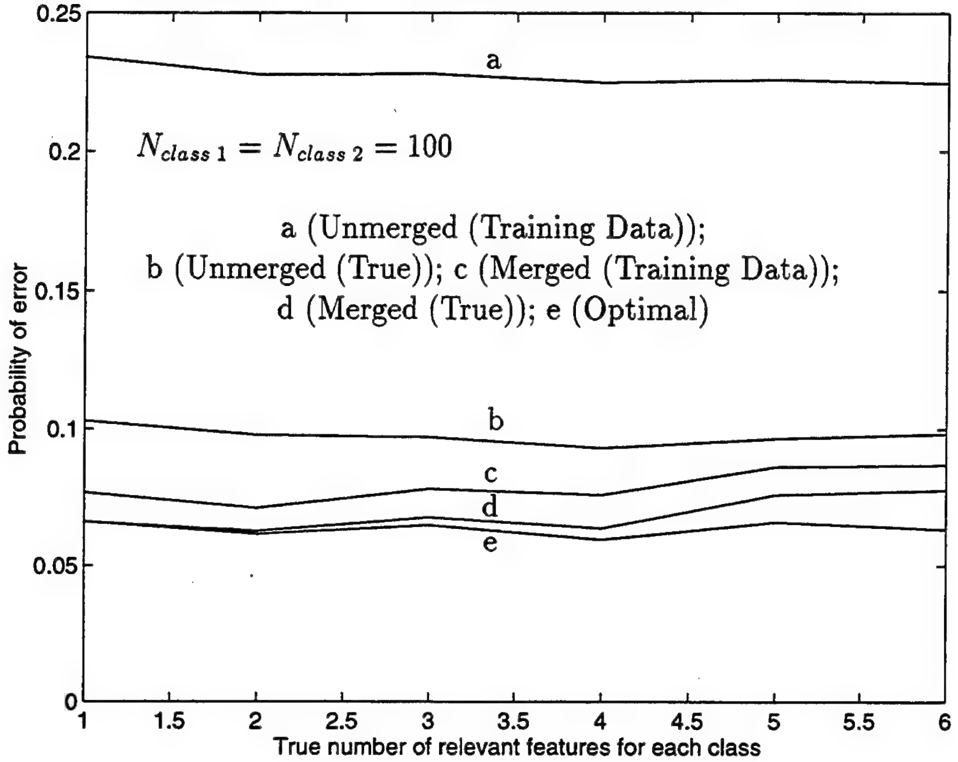


Figure 10: Performance results for the situation shown in Figure 8 with one hundred samples of training data for each class.

In Figures 10 and 11 the effects of increasing the training data size (i.e., by a factor of four or, $N_{class\ 1} = N_{class\ 2} = 100$) is demonstrated on the results shown in Figures 8 and 9. It can be seen, for example, that in Figure 10 the same general trend appears in these results except that now the error probabilities produced are better (i.e., smaller). This of course is directly related to the increase in training data size, which makes estimation of the underlying symbol probabilities more accurate. Additionally, note that by comparing Figure 8 to Figure 10, another effect of increasing the training data size is to diminish the variation of error probabilities with the number of relevant features. This then implies

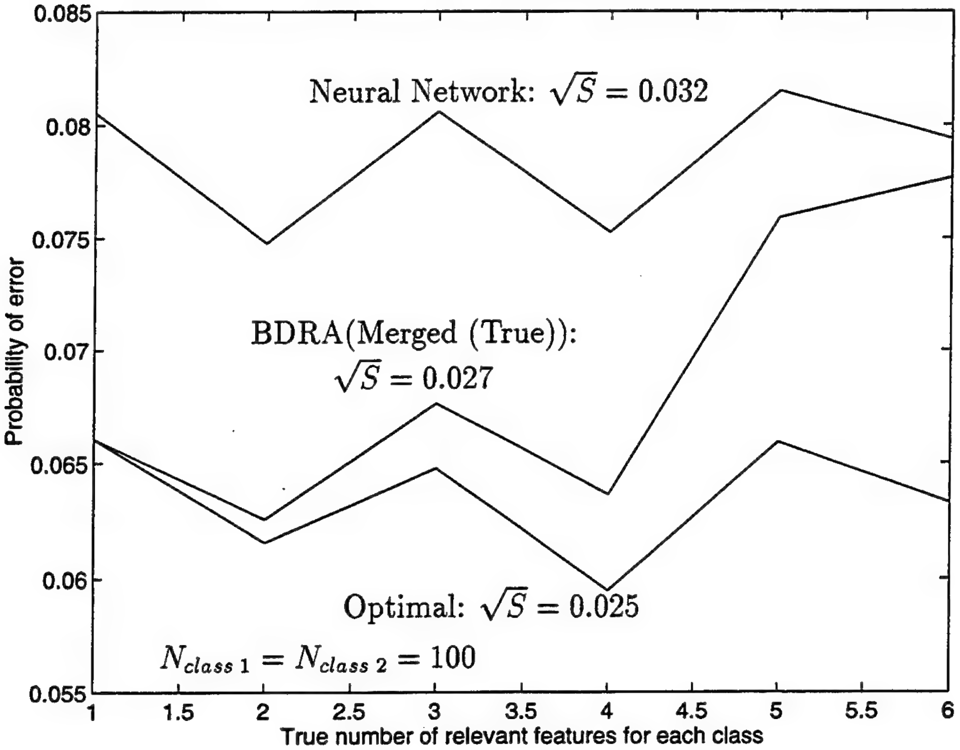


Figure 11: Performance comparison shown in Figure 9 with one hundred samples of training data for each class.

that more training data is helping to identify features which are useful to correct classification.

Continuing with Figure 11 as was done in Figure 9, the performance comparison to a neural network is now illustrated using the larger training data size of one hundred samples. Notice, the results labeled as Optimal and Merged (True) are again plotted from Figure 10, and their sample standard deviations are given by $\sqrt{S}$. Also, the evident zigzag pattern in the results is attributed to being an artifact of the way in which the true symbol probabilities are generated.

It can be seen in Figure 11 that the BDRA outperforms the neural network, as it did in Figure 9, for all possible numbers of relevant features. Also, it is apparent, and this was previously observed in Figure 10, that adding more training data has improved the performance of both classifiers, as their error probabilities are smaller than they were in Figure 9.

With these observations also notice in Figure 11, and this was not as apparent in Figure 9, that performance of the neural network appears to be less dependent on the number of relevant features than it is for the BDRA. That is, the BDRA has a tendency to be somewhat "eager" to reduce the training data. For an illustration of this consider Figure 12 below.

In Figure 12 below is shown the estimated number of relevant binary valued features by the BDRA, as a function of the actual number, for both training data sizes used previously. In other words, what appears in this figure is the number of features, on average, that remained after applying the BDRA to the training data. In general, completely accurate estimates would produce a straight line with a slope of unity. However, it can be seen that even with the larger training data size of one hundred samples the algorithm tends to over-reduce the data. The one exception to this is the obvious case of one true relevant feature. But, notice a trend appearing again here in that more training data helps in identifying those features which are most relevant to correct classification (i.e, the estimates of relevant feature number increase with training size).

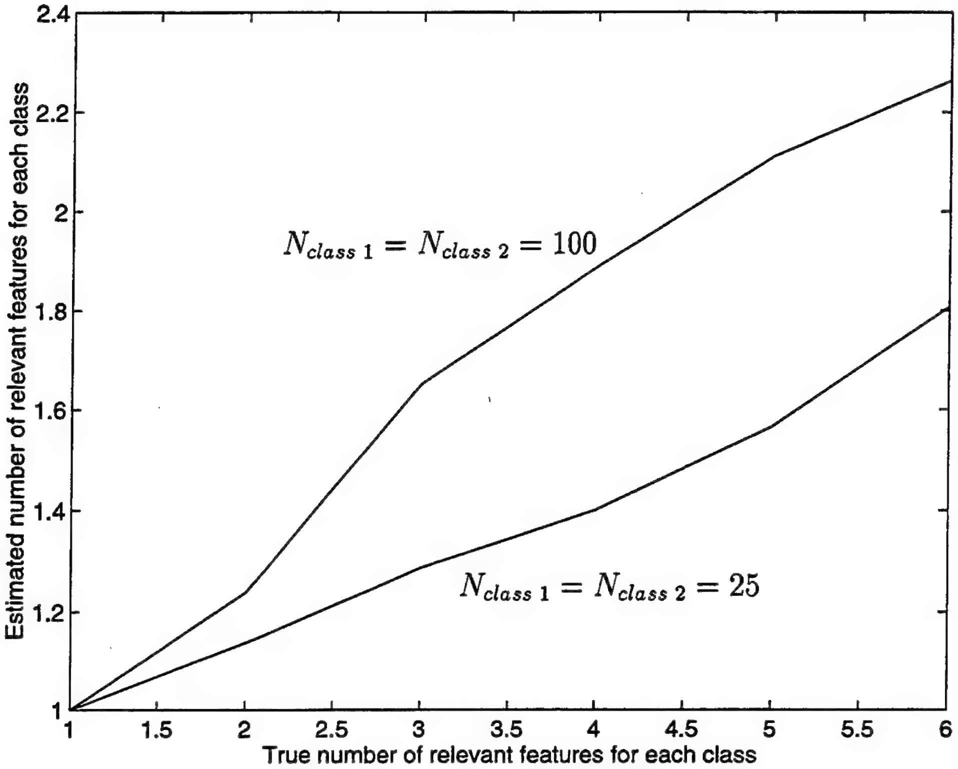


Figure 12: Estimated number of relevant binary valued features by the BDRA.

3.3.2 Performance at Reducing Ternary Valued Irrelevant Features

In Figures 13 and 14 below, the number of discrete levels is increased by one for each feature so that the training data consists of six ternary valued features. These figures are used to illustrate performance of the BDRA as the "curse of dimensionality," [2, 23], becomes more predominant in the data. Notice, these figures are similar to Figures 8 and 9 as they show error probabilities of the BDRA as a function of the number of relevant features for each class. But, it should also be pointed out that with ternary valued features (or those with a larger number of discrete levels) data reduction means all discrete levels of each

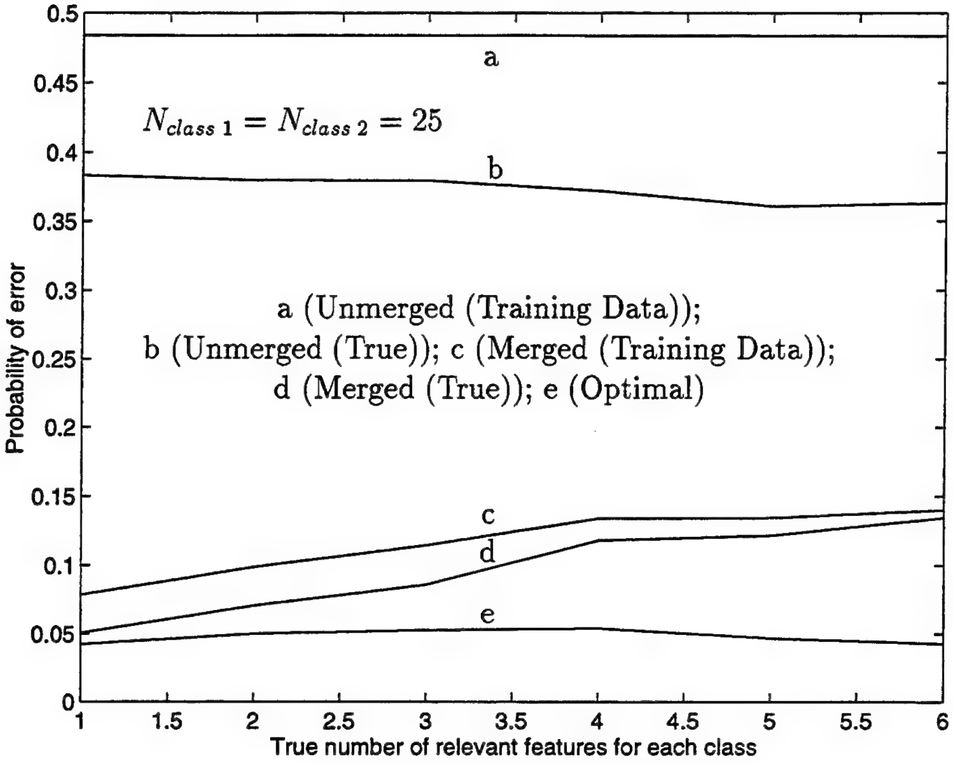


Figure 13: Performance of the BDRA with six ternary valued features, and twenty five samples of training data for each class.

feature are successively reduced one level at a time. That is, if reduced a ternary valued feature is first reduced to being binary valued, then, if reduced again it is eliminated. Further, results in these figures are based on twenty five samples of training data for each class.

The BDRA starts out in Figure 13 with a probability of error of near 0.5 (Unmerged (Training Data)), and then reduces this to less than 0.15 (Merged (Training Data)) for all numbers of true relevant features. Clearly, as compared to Figure 8 the effect of the curse of dimensionality can be seen in the high initial error probabilities. With this, under true statistics and for one relevant feature for each class, the BDRA is able to reduce the (Unmerged (True)) error probability from greater than 0.35 to near optimal (Merged (True)). In this way, the BDRA is an effective means of eliminating the curse of dimensionality, and this is addressed further in Chapter 6 where the BDRA is modified to improve its performance with high dimensional data. Also, and as expected from Figure 12, the relative loss in performance for the BDRA as the number of relevant features increases is more severe with ternary valued features. Again, this is attributed to the curse of dimensionality.

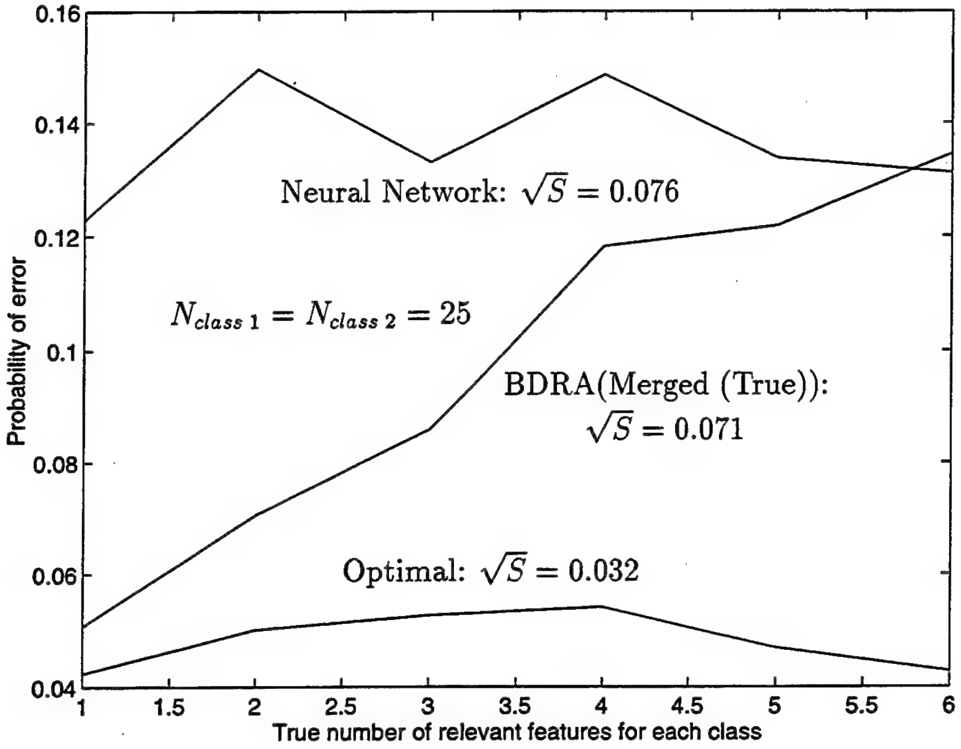


Figure 14: Performance comparison of the BDRA to a neural network with ternary valued features, and twenty five samples of training data for each class.

In Figure 14, performance of the BDRA is compared to a neural network. In this figure appears the Optimal and Merged (True) results from Figure 13, results for the neural network, and their respective average sample standard deviations, $\sqrt{S}$. Notice, the BDRA is shown to be overall superior to the neural network except when all of the features are relevant. Also, it can be seen that an additional effect of a small number of training data is the relatively large sample standard deviations of the BDRA and the neural network. However, as will be seen in Chapter 6 the negative effects of small sample size can be reduced (i.e., the BDRA can be improved) if the BDRA is modified to work only with those discrete symbols represented by the training data.

3.4 Summary

In this chapter, the Bayesian Data Reduction Algorithm (BDRA) was developed using the noninformative Dirichlet distribution as a prior on the symbol probabilities. Additionally, the overall performance of the BDRA was demonstrated, and it was shown to be superior to a neural network at reducing irrelevant binary and ternary valued features from the training data of each class. But, as expected, when both classes contain a complete set of relevant features, performance of the BDRA and a neural network are similar.

Chapter 4

The CBT and Mislabeled Training Data

4.1 Introduction

The subject of this chapter is to demonstrate performance of the CBT, using the average probability of error, $P(e)$, when the training data of each class are mislabeled. In this case, classification performance is shown to degrade when mislabeling exists in the training data, and this occurs with a severity that depends upon the mislabeling probabilities. Additionally, it is shown that as the mislabeling probabilities increase M^*, or the best quantization complexity related to the Hughes phenomenon (see, [23, 25, 34]), also increases. Notice, that even when the actual mislabeling probabilities are known by the CBT it is not possible to achieve the classification performance obtainable without mislabeling. However, it is also shown that the negative effect of mislabeling can be diminished, with more success for smaller mislabeling probabilities, if the BDRA of Chapter 3 is applied to the training data.

Observe, with the situation of interest the training data of each class are assumed to be made up of two parts: a correctly labeled part, and a mislabeled part. Specifically, the N_k (N_l) training data of class k (l) consists of N_{kk} (N_{ll}) correctly labeled observations occurring with probability $1 - \alpha_k$ ($1 - \alpha_l$), and a remaining N_{kl} (N_{lk}) mislabeled observations (i.e., belonging to the other class) occurring with probability α_k (α_l). Also, it is assumed that $N_{\mathbf{y}}$ unlabeled "test" data are observed. Thus, the problem addressed in this chapter is to illustrate, using $P(e)$, the effect that mislabeled training data has on classifying the unknown test data.

4.2 Classification With Mislabeled Training Data

4.2.1 Combined Multinomial Model

The combined multinomial for mislabeled training data is an extension of that shown in formula (1) of Chapter 2. Thus, the joint distribution for the frequency of occurrence of all training and test data with the test data, $\mathbf{y}$, a member of class k is given by

$$\begin{aligned}
&f\left(\mathbf{x}_{kk}, \mathbf{x}_{lk}, \mathbf{x}_{ll}, \mathbf{x}_{kl}, \mathbf{y} | \mathbf{p}_k, \mathbf{p}_l, H_k; \alpha_k, \alpha_l\right) \\
&\quad = N_{kk}!N_{lk}!N_{ll}!N_{kl}!N_{\mathbf{y}}! \prod_{i=1}^{M} \frac{p_{k,i}^{x_{kk,i}+x_{lk,i}+y_i} \, p_{l,i}^{x_{ll,i}+x_{kl,i}}}{x_{kk,i}!x_{lk,i}!x_{ll,i}!x_{kl,i}!y_i!} \\
&\quad \times \frac{N_k!}{N_{kk}!N_{kl}!} \left(\alpha_k\right)^{N_{kl}} \left(1 - \alpha_k\right)^{N_{kk}} \frac{N_l!}{N_{ll}!N_{lk}!} \left(\alpha_l\right)^{N_{lk}} \left(1 - \alpha_l\right)^{N_{ll}} \qquad (20)
\end{aligned}$$

where (as in previous chapters, k and l are exchangeable)

k, $l \in \{class\ 1, class\ 2\}$, and $k \neq l$;

H_k is the hypothesis defined as $\mathbf{p_y} = \mathbf{p}_k$;

M is the number of discrete symbols;

$x_{kk,i}$ is the number of occurrences of the i^{th} symbol in the correctly labeled training data for class k;

N_{kk} $\left\{N_{kk} = \sum_{i=1}^{M} x_{kk,i}\right\}$ is the number of correctly labeled training data for class k;

$x_{kl,i}$ is the number of occurrences of the i^{th} symbol in the mislabeled training data for class k, appearing with probability α_k and belonging to class l;

N_{kl} $\left\{N_{kl} = \sum_{i=1}^{M} x_{kl,i}\right\}$ is the number of mislabeled training data for class k;

$x_{k,i} = x_{kk,i} + x_{kl,i}$ is the number of occurrences of the i^{th} symbol in all training data for class k;

N_k $\left\{N_k = N_{kk} + N_{kl} = \sum_{i=1}^{M} x_{k,i}\right\}$ is the total number of training data for class k;

y_i is the number of occurrences of the i^{th} symbol in the test data;

$N_{\mathbf{y}}$ $\left\{N_{\mathbf{y}} = \sum_{i=1}^{M} y_i\right\}$ is the total number of test data;

$p_{k,i}$ $\left\{\sum_{i=1}^{M} p_{k,i} = 1\right\}$ is the probability of the i^{th} symbol for class k.

4.2.2 Combined Bayes Test (CBT)

The first step in developing the CBT for mislabeled training data is to apply the Dirichlet of formula (2), Chapter 2, to the formula of (20) under each class

k and l, and then integrate with respect to $\mathbf{p}_k$ and $\mathbf{p}_l$ over the *positive unit-hyperplane* resulting in

$$
\begin{aligned}
&f\left(\mathbf{x}_{kk}, \mathbf{x}_{lk}, \mathbf{x}_{ll}, \mathbf{x}_{kl}, \mathbf{y} | H_k; \alpha_k, \alpha_l\right) \\
&\quad = \frac{\left[(M-1)!\right]^2 N_{kk}! N_{lk}! N_{ll}! N_{kl}! N_{\mathbf{y}}!}{\left(N_{kk} + N_{lk} + N_{\mathbf{y}} + M - 1\right)! \left(N_{ll} + N_{kl} + M - 1\right)!} \\
&\quad \times \prod_{i=1}^{M} \frac{\left(x_{kk,i} + x_{lk,i} + y_i\right)! \left(x_{ll,i} + x_{kl,i}\right)!}{x_{kk,i}! x_{lk,i}! x_{ll,i}! x_{kl,i}! y_i!} \\
&\quad \times \frac{N_k!}{N_{kk}! N_{kl}!} \left(\alpha_k\right)^{N_{kl}} \left(1 - \alpha_k\right)^{N_{kk}} \frac{N_l!}{N_{ll}! N_{lk}!} \left(\alpha_l\right)^{N_{lk}} \left(1 - \alpha_l\right)^{N_{ll}}. \qquad (21)
\end{aligned}
$$

Continuing, formula (21) is now expressed in terms of the complete training data vectors, $\mathbf{x}_k$ and $\mathbf{x}_l$. This is accomplished by substituting the definitions $\mathbf{x}_{kk} = \mathbf{x}_k - \mathbf{x}_{kl}$ and $\mathbf{x}_{ll} = \mathbf{x}_l - \mathbf{x}_{lk}$ into formula (21), followed by summing over all possible arrangements of mislabeled training data vectors, yielding

$$
\begin{aligned}
&f\left(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y} | H_k; \alpha_k, \alpha_l\right) \\
&\quad = \sum_{\mathbf{x}_{kl}=\vec{0}}^{\mathbf{x}_k} \sum_{\mathbf{x}_{lk}=\vec{0}}^{\mathbf{x}_l} f\left(\mathbf{x}_k - \mathbf{x}_{kl}, \mathbf{x}_{lk}, \mathbf{x}_l - \mathbf{x}_{lk}, \mathbf{x}_{kl}, \mathbf{y} | H_k; \alpha_k, \alpha_l\right). \qquad (22)
\end{aligned}
$$

Using this result, the CBT is then given by the ratio of (22) to its analogous formula under class l (i.e., conditioned on H_l), and it appears as

$$
\frac{f\left(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y} | H_k; \alpha_k, \alpha_l\right)}{f\left(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y} | H_l; \alpha_k, \alpha_l\right)} \underset{H_l}{\overset{H_k}{\gtrless}} \tau \qquad (23)
$$

where, for minimizing the probability of error the decision threshold τ is equal to $P\left(H_l\right) / P\left(H_k\right)$.

4.2.3 Probability of Error

Letting $z_k = f\left(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y} | H_k; \alpha_k, \alpha_l\right)$ (see formula (22) above), the average probability of error for the CBT is defined as

$$P\left(e\right) = P\left(H_k\right) P\left(z_k \leq \tau z_l \mid H_k\right) + P\left(H_l\right) P\left(z_k > \tau z_l \mid H_l\right). \tag{24}$$

It is necessary to show the first term of (24) only as the second term is similar except for conditioning on H_l. Thus, ignoring $P\left(H_k\right)$, the first term of (24) is given by

$$P\left(z_k \leq \tau z_l \mid H_k\right) = \sum_{\mathbf{y}} \sum_{\mathbf{x}_k} \sum_{\mathbf{x}_l} \mathcal{I}_{\{z_k \leq \tau z_l\}} f\left(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y} | H_k; \alpha_k, \alpha_l\right) \tag{25}$$

where $f\left(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y} | H_k; \alpha_k, \alpha_l\right)$ was defined in formula (22) above.

Before illustrating results for the CBT notice that analogous to formula (9) of Chapter 2 formula (25) can also be rewritten for $N_{\mathbf{y}} = 1$, and this requires that $f\left(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y} | H_k; \alpha_k, \alpha_l\right)$ be given by (see the accompanying text of formula (8) in Chapter 2 for notational descriptions)

$$\begin{aligned}
&\sum_{\mathbf{x}_{kn}} \sum_{\mathbf{x}_{ln}} f\left(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y} | H_k; \alpha_k, \alpha_l\right) \\
&= \sum_{\mathbf{x}_{kn}} \sum_{\mathbf{x}_{ln}} f\left(x_{k,ir}, x_{l,ir}, y_{ir} = 1 | H_k; \alpha_k, \alpha_l\right) = \sum_{x_{kl,ir}=0}^{x_{k,ir}} \sum_{x_{lk,ir}=0}^{x_{l,ir}} \sum_{\sum \mathbf{x}_{kkn}=0}^{\sum \mathbf{x}_{kn}} \sum_{\sum \mathbf{x}_{lln}=0}^{\sum \mathbf{x}_{ln}} \\
&\times \frac{\left[(M-1)!\right]^2 N_{kk}! N_{lk}! N_{ll}! N_{kl}!}{\left(N_{ll} + N_{kl} + M - 1\right)! \left(N_{kk} + N_{lk} + M\right)!} \frac{\left(x_{kk,ir} + x_{lk,ir} + 1\right)! \left(x_{ll,ir} + x_{kl,ir}\right)!}{x_{kk,ir}! x_{lk,ir}! x_{ll,ir}! x_{kl,ir}!}
\end{aligned}$$

$$\times \binom{\sum \mathbf{x}_{kkn} + \sum \mathbf{x}_{lkn} + M - 2}{\sum \mathbf{x}_{kkn} + \sum \mathbf{x}_{lkn}} \binom{\sum \mathbf{x}_{lln} + \sum \mathbf{x}_{kln} + M - 2}{\sum \mathbf{x}_{lln} + \sum \mathbf{x}_{kln}}$$

$$\times \frac{x_{k,ir}!}{x_{kk,ir}! x_{kl,ir}!} (\alpha_k)^{N_{kl}} (1 - \alpha_k)^{N_{kk}} \frac{x_{l,ir}!}{x_{ll,ir}! x_{lk,ir}!} (\alpha_l)^{N_{lk}} (1 - \alpha_l)^{N_{ll}} . \quad (26)$$

Notice, as with formula (9) of Chapter 2, because of symmetry in the Dirichlet distribution formula (26) is equal for all "ir" in $\{1, 2, ..., M\}$. With this, when $N_{\mathbf{y}} = 1$ the summation over $\mathbf{y}$ in formula (25) involves a sum of the same M terms. Also, the notation $\sum \mathbf{x}_{kkn}$ means the sum of all correctly labeled training data under k that are not represented by the same discrete symbol as the test observation. Further, $\binom{\sum \mathbf{x}_{kkn} + \sum \mathbf{x}_{lkn} + M - 2}{\sum \mathbf{x}_{kkn} + \sum \mathbf{x}_{lkn}}$ means the number of ways $\sum \mathbf{x}_{kkn} + \sum \mathbf{x}_{lkn}$ training data can be arranged amongst $M - 1$ discrete symbols.

Now, for larger values of $N_{\mathbf{y}}$ formula (26) can be straightforwardly extended. For example, $N_{\mathbf{y}} = 2$ requires the sum of two terms. That is, formula (26) must be extended to obtain $f(x_{k,i}, x_{l,i}, y_i = 2 | H_k; \alpha_k, \alpha_l)$, and the formula it is to be summed with, $f(x_{k,i}, x_{k,j}, x_{l,i}, x_{l,j}, y_i = 1, y_j = 1 | H_k; \alpha_k, \alpha_l)$. In any case, the benefit of using formula (26) is that it simplifies the necessary computations of formula (24), and the results shown below are based on this simplification.

4.3 Results

Figure 15 below contains an average probability of error curve, $P(e)$ (plotted as a function of the number of discrete symbols, M), for the CBT given the true mislabeling probabilities are given by, respectively, $\alpha_k = \alpha_l = 0.0$, 0.05, 0.15,

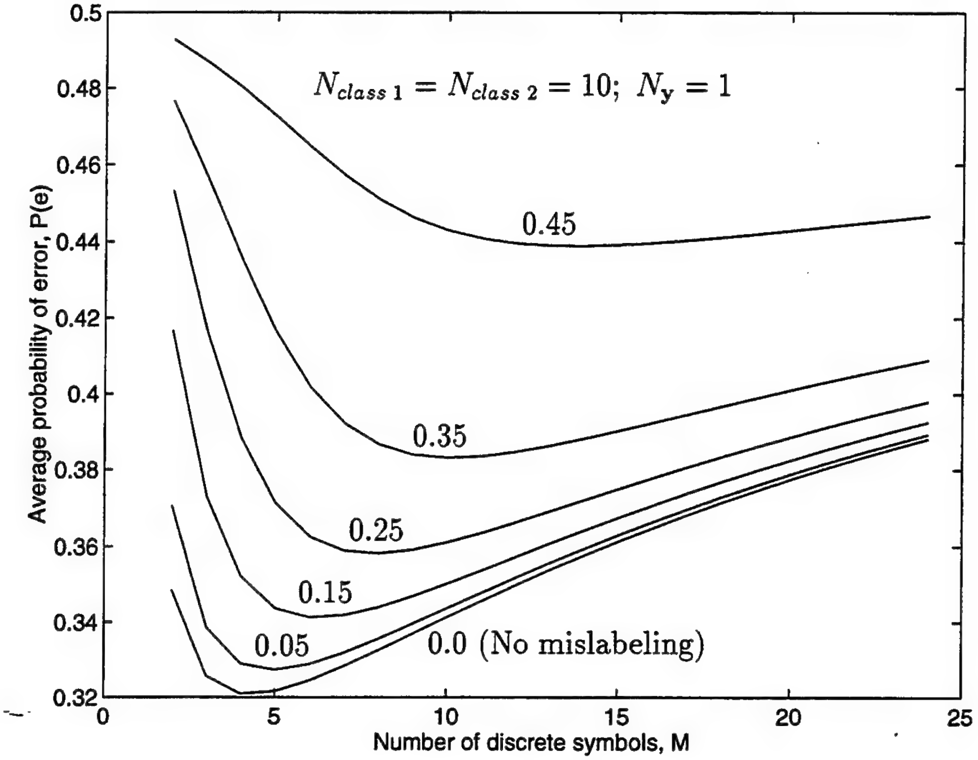


Figure 15: $P(e)$ with various mislabeling probabilities.

0.25, 0.35, and 0.45. Notice that results are based on ten samples of training data for each class, and one observation of test data. Additionally, the decision threshold $\tau = 1$. In all cases of Figure 15, observe that $P(e)$ starts out decreasing with increasing M and is minimum at the point called M^*. For example, when there is no mislabeling in the training data $M^* = 4$, and this case previously appeared in Figure 4 of Chapter 2. Then, for M greater than M^* $P(e)$ steadily increases. This dependence of $P(e)$ on M was addressed in Chapter 2 and it reflects the fact that given a fixed amount of training and test data a prior quantizing complexity exists which yields, on average, the "best" classification performance [34]. However, as the mislabeling probabilities are fixed with larger values overall performance begins to degrade in that $P(e)$ increases. Also, it can be seen that accompanying this degradation in performance is an increase in M^*. That is, for the mislabeling probabilities in Figure 15 given by 0.0, 0.05, 0.15, 0.25, 0.35, and 0.45, M^* has the respective values of 4, 5, 6, 8, 10, and 12. Notice, it can be seen that even when $\alpha_k = \alpha_l = 0.45$, a best quantization complexity exists. Intuitively, an increase in the mislabeling probabilities causes the classes to become similar, so that for best classification performance more information (i.e., a finer quantization) is required.

With these findings, it was found that if the mislabeling probabilities assumed for the training data (i.e., for the z_k and z_l of formula (24)) take on any values within the range, $0 \leq \alpha_k = \alpha_l < 0.5$, identical results are produced for all cases in Figure 15. In other words, when testing it is does not matter if the CBT of formula (23) contains the true mislabeling probabilities as long as they are not assumed to be 0.5 or higher (which would indicate a CBT that is testing as if most of the training data of each class is more likely to belong the other class). This aspect of the CBT's performance is attributed to the averaging which occurs in formula (22) over all possible orderings of the mislabeled training data, coupled with placement of the uniform (i.e., Dirichlet) prior on the symbol probabilities.

In Figure 16 below, results from Figure 15 are repeated (i.e., $N_{\mathbf{y}} = 1$) for the mislabeling probabilities given by $\alpha_k = \alpha_l = 0.0, 0.05.$ 0.15, and 0.25. Additionally, also shown (lower curves with an *) for the same mislabeling probabilities

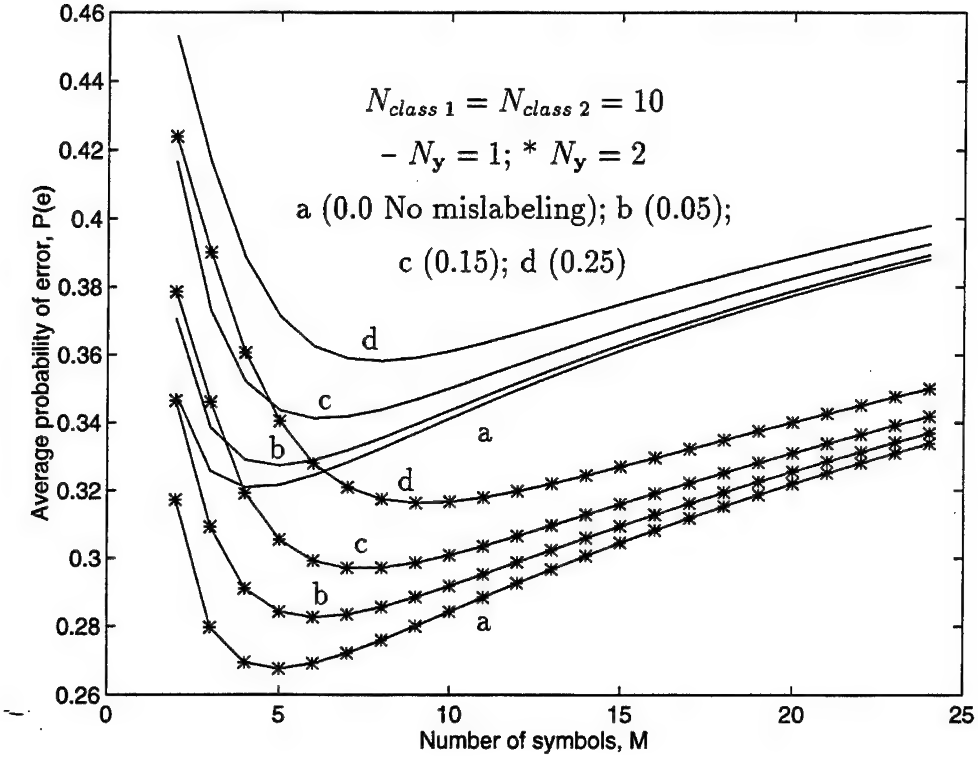


Figure 16: $P(e)$ with more test observations.

is the case involving two observations of test data (i.e., $N_y = 2$). It can be seen in this figure that when $N_y = 2$ performance improves for a given mislabeling probability as $P(e)$ is reduced (see Figure 4 of Chapter 2). With this, observe that as compared to the $N_y = 1$ case, increasing the number of test observations to $N_y = 2$ causes all associated values of M^*, where performance is best, to increase by one. That is, for the mislabeling probabilities given by 0.0, 0.05, 0.15, and 0.25, and when $N_y = 2$, M^* has the respective values of 5, 6, 7, and 9. Accompanying this increase in M^* is the associated increase in $P(e)$. However, it is apparent that the increase in $P(e)$ is relatively worse when $N_y = 2$ (i.e., the

$P(e)$ curves are further apart). The reason this occurs is that although a greater number of test observations improves the estimation capability of the CBT, there also is more of a likelihood that a test observation will be of the same value as a mislabeled training datum.

4.4 Applying the BDRA to Mislabeled Training Data

In this section the Bayesian Data Reduction Algorithm (BDRA) is applied to mislabeled training data. As used here, the BDRA demonstrates the degree to which the negative effect of mislabeling can be diminished by employing a suboptimal algorithm to train on the data. Performance of the BDRA is described in Chapter 3 at classifying, and reducing, feature vectors containing binary and ternary valued features. In these cases, the BDRA was shown to be superior to a neural network. Here, the BDRA is applied to feature vectors consisting of six binary valued features (i.e., $M = 64$), which are also mislabeled in the training data of each class according to the probabilities shown in Figure 15.

Recall, the BDRA works by reducing the quantization complexity, M, of the training data to a level which minimizes the average conditional probability of error, $P(e \mid X)$ (where X represents the entire collection of training data from all classes). It was shown in Chapter 3 that the formula for $P(e \mid X)$ is a fundamental component of the BDRA, and repeated here it is given by (note, the notational descriptions of formula (20) apply to formula (27))

$$
\begin{aligned}
P\left(e \mid X\right) &= P\left(e \mid \mathbf{x}_k, \mathbf{x}_l\right) \\
&= \sum_{\mathbf{y}} \sum_{\mathbf{x}_k, \mathbf{x}_l} P\left(H_k\right) \mathcal{I}_{\{z_k \leq z_l\}} f\left(\mathbf{y} | \mathbf{x}_k, H_k\right) + P\left(H_l\right) \mathcal{I}_{\{z_k > z_l\}} f\left(\mathbf{y} | \mathbf{x}_l, H_l\right) \qquad (27)
\end{aligned}
$$

where in the cases considered involving only one observation of test data (i.e., $N_{\mathbf{y}} = 1$) $z_k = f\left(\mathbf{y}|\mathbf{x}_k, H_k\right) = \left(x_{k,i} + 1\right) / \left(N_k + M\right)$.

For binary valued feature vectors the six iterative steps of the BDRA are also repeated here from Chapter 3.

1. Using the initial training data with quantization complexity M (i.e., $M = 2^{N_f}$, where N_f is the number of features), use formula (24) to compute $P\left(e \mid X; M\right)$.

2. Beginning with the first feature (selection is arbitrary), remove this feature from each class by summing (i.e., merging) the numbers of occurrences of those discrete symbols that correspond to its removal (i.e., for all classes simultaneously merge those quantized symbols containing a binary zero for that reduced feature with those containing a binary one).

3. Use the newly merged training data (X') and the new quantization complexity $(M' = 2^{N_f - 1})$, and compute $P(e \mid X'; M')$.

4. Repeat items two and three for all N_f features.

5. From item four select the minimum of all computed $P(e \mid X'; M')$ (ties are broken arbitrarily), and choose this as the new training data configuration

for each class (this corresponds to permanently removing the associated feature).

6. Repeat items two through five until the probability of error does not decrease any further, or $M' = 2$, and this defines the new quantization complexity.

4.5 Results Using the BDRA

In Figure 17 below, performance of the BDRA is shown when the training data are mislabeled according to the probabilities specified in Figure 15. The results in this figure are based on an average of one hundred independent trials. At each trial, a set of $M = 64$ true symbol probabilities, consisting of six independent bit probability pairs, were generated for both classes using Gaussian mixture distributions (see Chapter 3). Additionally, results appear for two training data sizes of twenty five and one hundred samples, which were randomly generated at each trial from the true symbol probabilities.

Observe in Figure 17 that P(e) appears as a function of the mislabeling probabilities both with and without applying the BDRA to the training data, and for the optimal test. Note that results shown for the BDRA were obtained by using its trained test statistic with the actual symbol probabilities. Also, optimal results are based on the test which knows all true symbol probabilities, and there is no mislabeling of the training data. With this, it can be seen that the optimal

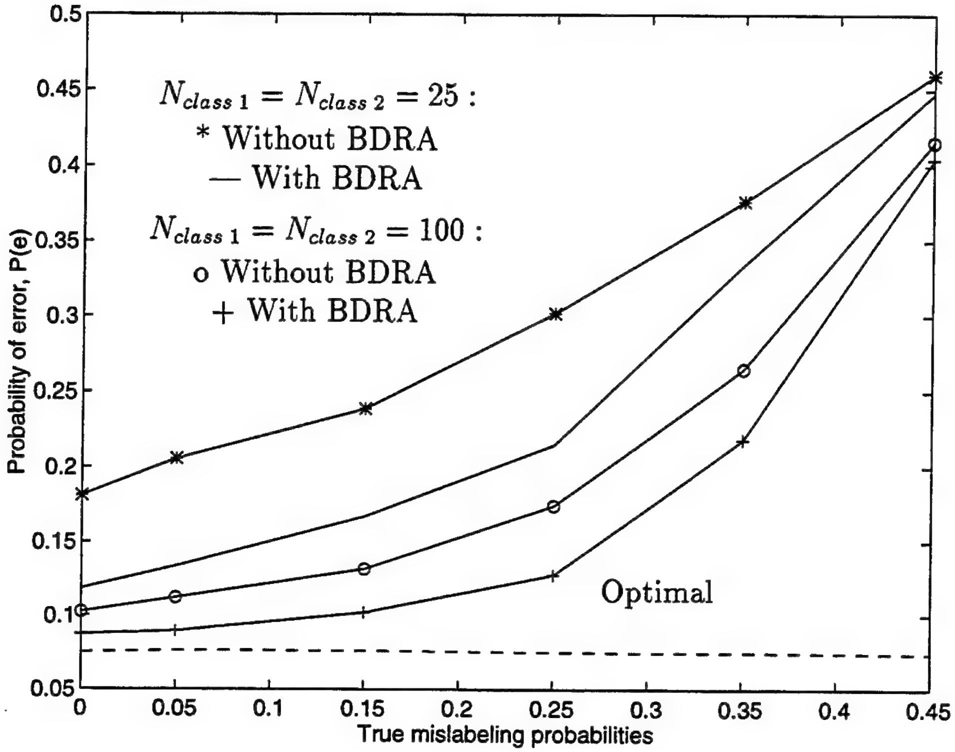


Figure 17: Performance of the BDRA with the mislabeling probabilities of Figure 15.

error probabilities are relatively constant at 0.075, and this is due to them having been constrained to be ≥ 0.05 and ≤ 0.1. This constraint on the optimal error probabilities was possible because the true symbol probabilities were created with Gaussian mixture distributions (see Chapter 3). Notice in Figure 17 that in all cases performance degrades with the severity of the mislabeling probabilities, which is analogous to the results in Figure 15. However, for both training data sizes the BDRA is successful at improving overall classification performance (relatively less improvement, that is, less data reduction, occurs with more training data as the probability estimates are more accurate). But, in all cases it appears that the improvement diminishes rapidly as the mislabeling probabilities approach 0.5. On the other hand, with one hundred samples of training data and mislabeling probabilities of less than 0.1, performance is relatively close to optimal.

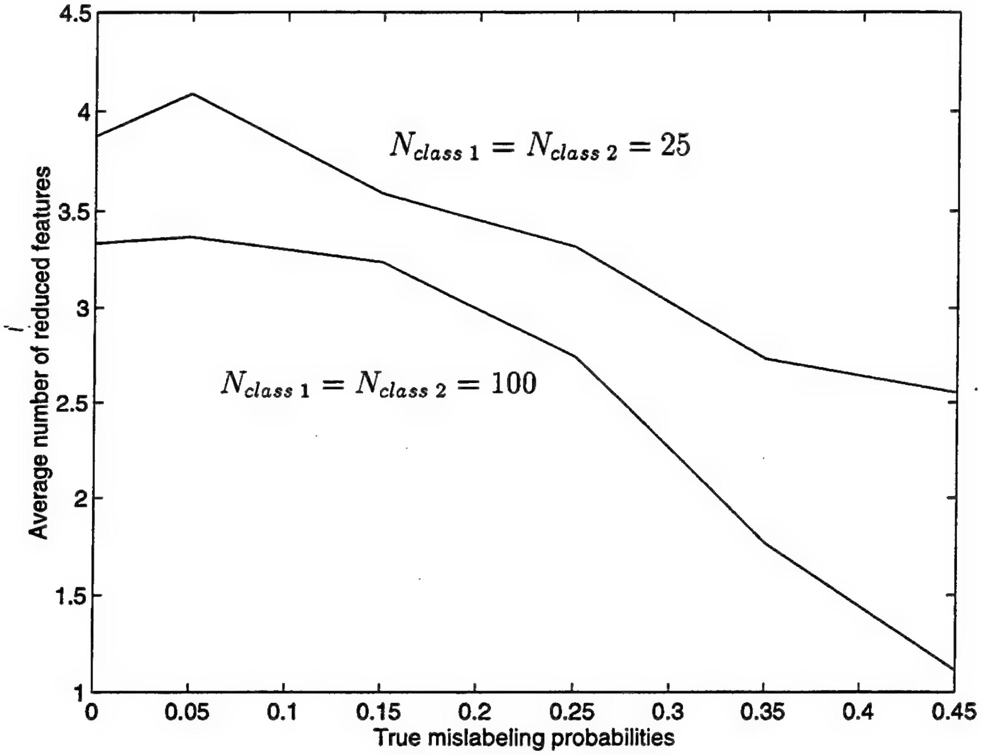


Figure 18: Average number of relevant features reduced, out of a total of six, from the training data of each class.

Figure 18 shows the average number of relevant features reduced, out of a total of six, from the training data of each class by the BDRA as a function of the true mislabeling probabilities. In this figure, results appear for those training data sizes shown in Figure 17, and are based on an average of one hundred independent trials. As expected from Figure 15, overall it can be seen that the average number of features reduced (eliminated) from the training data of each class becomes less as the mislabeling probabilities increase (the increase in the number of features associated with a mislabeling probability of 0.05 is attributed to using only one hundred independent trials to obtain the results). Also, consistent with the results of Figure 17, the BDRA appears to reduce a larger number of features when there is less training data, and this is caused by relatively more uncertainty associated with the symbol probability estimates.

4.6 Summary

The subject of this chapter has been the effect that mislabeled training data has on classification performance, given there is no knowledge of the underlying discrete symbol probabilities. In general, it was shown that as the mislabeling probabilities increase, both the average probability of error and the optimum quantization complexity, M^*, increase. Additionally, it was found that $P(e)$ can be reduced if the number of test observations is increased to $N_{\mathbf{y}} = 2$. However, the relative performance degradation with mislabeling present is relatively larger then it is when $N_{\mathbf{y}} = 1$, and this is due to an increased likelihood of the test data matching the mislabeled training data. Further, the BDRA was applied to training data corrupted by mislabeling, and results indicate that classification performance can be improved if the mislabeling probabilities are not too severe. But, the relative amount of improvement decreases with training data size as the symbol probability estimates become more accurate.

Chapter 5

The BDRA and Missing Features

5.1 Introduction

In this chapter, the BDRA is used to classify test observations given that the training data of each class is missing feature information. Observe, by missing features it is meant that each of the N_k feature vectors of the training data for class k are assumed to be made up of either or both of the following two observation types: features which are represented by discrete values, and missing features which have no values. For example, with three binary features a possible feature vector that is missing a single feature might appear as $(1, 1, \mathrm{x})$, where x represents the missing value. In this case, x can have the value of 0 or 1 so that this feature vector has a cardinality of two. Notice, the missing features are assumed to appear according to an unknown probability distribution. But, when simulating training data with missing features a uniform random variable is used to control their frequency of occurrence.

Now, in the BDRA the missing feature information is modeled using two different approaches. With the first of these approaches, the Dirichlet prior is extended to accommodate missing features in the natural way. That is, each missing feature is assumed to be uniformly distributed over its range of values. In the second approach, the number of discrete levels for each feature is increased by one so that all missing values for that feature are assigned to the same level. To illustrate performance, the BDRA is compared to a neural network at classifying binary valued feature vectors, with the missing features appearing randomly in the training data of each class. Note, in the case of the neural network each missing feature is assigned to an additional value as is done for the second approach of the BDRA. In general, simulation results with six binary valued features reveals that both approaches to modeling missing features in the BDRA perform similarly, and they also are both superior to the neural network.

5.2 The BDRA Extended for Missing Features

5.2.1 Method 1

In the first method, development of the BDRA for the missing features problem relies on the underlying assumption that for each missing feature vector its cardinality of values (i.e., this is all possible discrete symbols the feature vector can take on if all possible arrangements of values are substituted in for the missing features) are uniformly distributed. Given that, the probability of observing, for the k^{th} class, a specific arrangement of discrete symbols in the training data

($\mathbf{w}_k$) and a single test observation of type i ($y_i = 1$) is defined as (extensions to $N_\mathbf{y} > 1$ are shown below to be straightforward)

$$f\left(\mathbf{w}_k, y_i = 1 \mid \mathbf{p}_k; H_k\right) = p_{k,i} \prod_{j=1}^{N_k} \left[\sum_{n \in w_{k,j}} p_{k,n} \right] \tag{28}$$

where (also see the notational definitions of Section 3.2 of Chapter 3) $w_{k,j}$ is a single observation of a feature vector in the training data of class k. Notice, without missing features $w_{k,j}$ is a single observation of a symbol of type i, and with missing features $w_{k,j}$ is one of $\mid w_{k,j} \mid$ possible symbols (i.e., $\mid w_{k,j} \mid$ is the cardinality of $w_{k,j}$ or, the number of possible symbols it can take on after substituting in all arrangements of missing feature values).

Notice, the formula of (28) above represents the sum of $\prod_{j=1}^{N_k} \mid w_{k,j} \mid$ terms (or all possible arrangements of the training data given the missing feature information) each having the form

$$\left(p_{k,1}^{c_1}, \ldots, p_{k,i}^{b+1}, \ldots, p_{k,M}^{c_M}\right) \tag{29}$$

where, for example, $b+1$ is the number of $p_{k,i}$'s in the product.

Now, the multinomial coefficients, $\left(\frac{N_k!}{c_1!, \ldots, b!, \ldots, c_M!}\right)$, are multiplied by the product in (29) and the uniform Dirichlet distribution of formula (2) in Chapter 2. Then, integration is carried out over the unit simplex producing the result

$$(b+1)\,\frac{N_k!(M-1)!}{(N_k+M)!}. \tag{30}$$

Thus, applying the above result to all terms in formula (28) yields

$$f\left(\mathbf{w}_k, y_i = 1 \mid H_k\right) = \sum_b C_b \left(b+1\right) \frac{N_k!(M-1)!}{(N_k+M)!} \tag{31}$$

in which C_b is the number of terms in each product of formula (28) containing $(b+1)$ $p_{k,i}$'s.

From here, a fictitious Bernoulli random variable Y_j is defined such that $\sum_{j=1}^{N_k} Y_j = b$. Thus, if S_i is defined as the event of being all those $w_{k,j}$ that can take on symbol i, and with each discrete symbol that each $w_{k,j}$ can take on being equally likely, the following probabilities are straightforward to see,

$$Pr\left(Y_j = 1\right) = \left\{ \begin{array}{cc} 0 & j \in S_i^c \\ \frac{1}{|w_{k,j}|} & j \in S_i \end{array} \right\}. \tag{32}$$

Now, using (32) and the definition of C_b produces the formula

$$\left(\prod_{j=1}^{N_k} \mid w_{k,j} \mid \right)^{-1} C_b = Pr\left(\sum_{j=1}^{N_k} Y_j = b \right). \tag{33}$$

Substituting formula (33) in formula (31), and summing over all possible values of b results in

$$f\left(\mathbf{w}_k, y_i = 1 \mid H_k\right) = \left(\prod_{j=1}^{N_k} \mid w_{k,j} \mid \right) \frac{N_k!(M-1)!}{(N_k+M)!} \left(1 + \sum_{j \in S_i} \frac{1}{\mid w_{k,j} \mid} \right). \tag{34}$$

Given the result above, without a test observation formula (34) becomes

$$f\left(\mathbf{w}_k \mid H_k\right) = \left(\prod_{j=1}^{N_k} \mid w_{k,j} \mid\right) \frac{N_k!(M-1)!}{(N_k+M-1)!}. \tag{35}$$

Therefore, for missing features the desired conditional distribution for the BDRA, or the z_k of formula (18) in Chapter 3, is produced by dividing formula (34) by formula (33), resulting in

$$z_k = f\left(y_i = 1 \mid \mathbf{w}_k, H_k\right) = \frac{\left(1 + \sum_{j \in S_i} \frac{1}{|w_{k,j}|}\right)}{N_k + M}. \tag{36}$$

Now, based on formula (17) of Chapter 3, it is straightforward to see that formula (36) for $N_\mathbf{y} \geq 1$ appears as

$$f\left(\mathbf{y}|\mathbf{w}_k, H_k\right) = \frac{(N_k+M-1)!\,(N_\mathbf{y})!}{(N_k+N_\mathbf{y}+M-1)!} \prod_{i=1}^{M} \frac{\left(\sum_{j \in S_i} \frac{1}{|w_{k,j}|} + y_i\right)!}{\left(\sum_{j \in S_i} \frac{1}{|w_{k,j}|}\right)!\,(y_i)!}. \tag{37}$$

The final step with this method is to develop a formula for the situation of missing features in both the training and test data, and this is given by

$$f\left(\mathbf{y}|\mathbf{w}_k, H_k\right) = \frac{(N_k+M-1)!\,(N_\mathbf{y})!}{(N_k+N_\mathbf{y}+M-1)!} \prod_{i=1}^{M} \frac{\left(\sum_{j \in S_i} \frac{1}{|w_{k,j}|} + \sum_{j \in S_{y,i}} \frac{1}{|w_{y,j}|}\right)!}{\left(\sum_{j \in S_i} \frac{1}{|w_{k,j}|}\right)!\left(\sum_{j \in S_{y,i}} \frac{1}{|w_{y,j}|}\right)!}. \tag{38}$$

where (see formula (28))

$w_{y,j}$ is a single observation of a feature vector in the test data;

$\mid w_{y,j} \mid$ is the cardinality of $w_{y,j}$;

$S_{y,i}$ is defined as the event of being all those $w_{y,j}$ that can take on symbol i.

5.2.2 Method 2

The BDRA is also extended for the missing features problem using a second method, which requires no additional probabilistic development. The basic idea of this method is to increase the number of discrete levels by one for each feature that has missing values (this actually represents one of several possible filling in type methods used with neural networks, [7]). Thus, the additional level created for each feature is used to represent each missing value. For example, for six binary valued features, and if it is known that any of these features can be missing from either the training or test data, then the initial quantization complexity is increased from $M = 64$ to $M = 729$. That is, each feature is ternary valued instead of binary valued. In the next section, results are shown for both of these methods.

5.3 Results

Figure 19 below shows error probabilities for the BDRA (using Method 1), a neural network, and the optimal test as a function of the true number of relevant features for each class (each class contains a total of six binary valued features). Note, that in this case optimal error probabilities have been constrained to be ≥ 0.05 and ≤ 0.1 (for more on this and the notation used here see Figures 8 through 14 of Chapter 3, and the accompanying text). Additionally, there are twenty five samples of training data for each class, and results are based on an average of one hundred independent trials. For the case shown in Figure

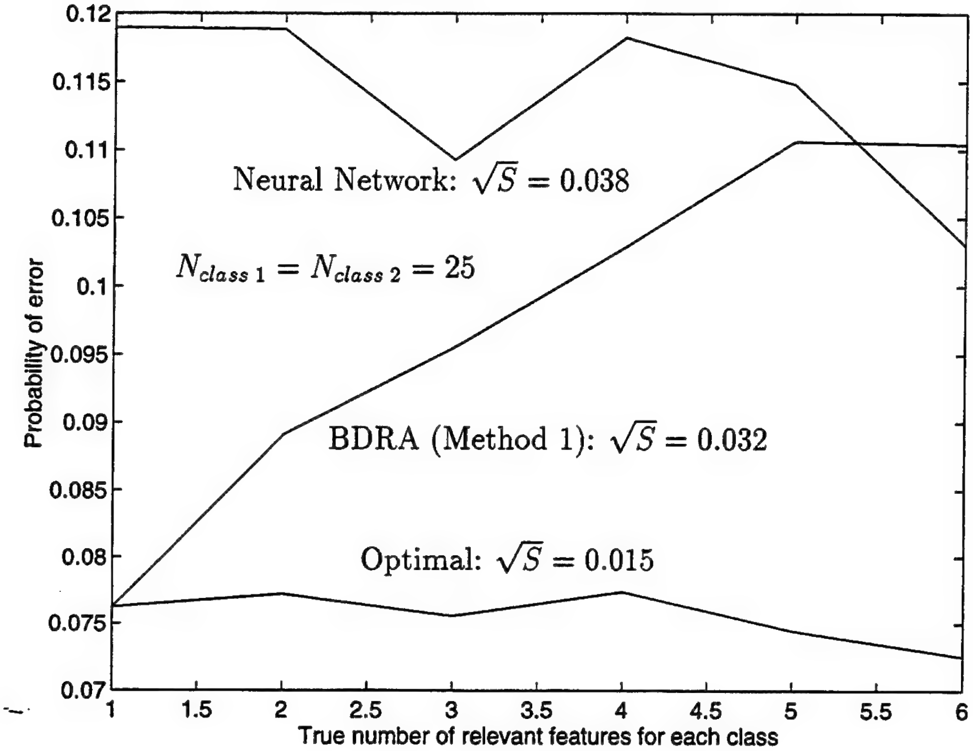


Figure 19: Performance comparison of the BDRA to a neural network when a random number of missing features occurs with a probability of 0.15.

19, a 0.15 probability exists for each class that up to three randomly selected features will be missing from the feature vectors in the training data (no missing features appear in the test data). Also, the neural network is trained on ternary valued feature vectors where a third discrete level is used for each missing feature. Alternatively, the neural network was also trained by substituting the average of the known feature values in for each missing feature, and this produced no substantial change in performance. Observe in Figure 19 that the BDRA is superior to the neural network by achieving an overall lower probability of error. However, the sample standard deviation, $\sqrt{S}$, of both schemes is similar, and this is partly due to a tighter constraint on the optimal error. Additionally, and as previously mentioned in Chapter 3, the BDRA performs best when the number of relevant features is minimum, whereas the opposite is true for the neural network. Further, and not shown here, the performance of both classifiers approach each other as the training data size increases, and this is expected based on the results of Chapter 3 (see Figure 11).

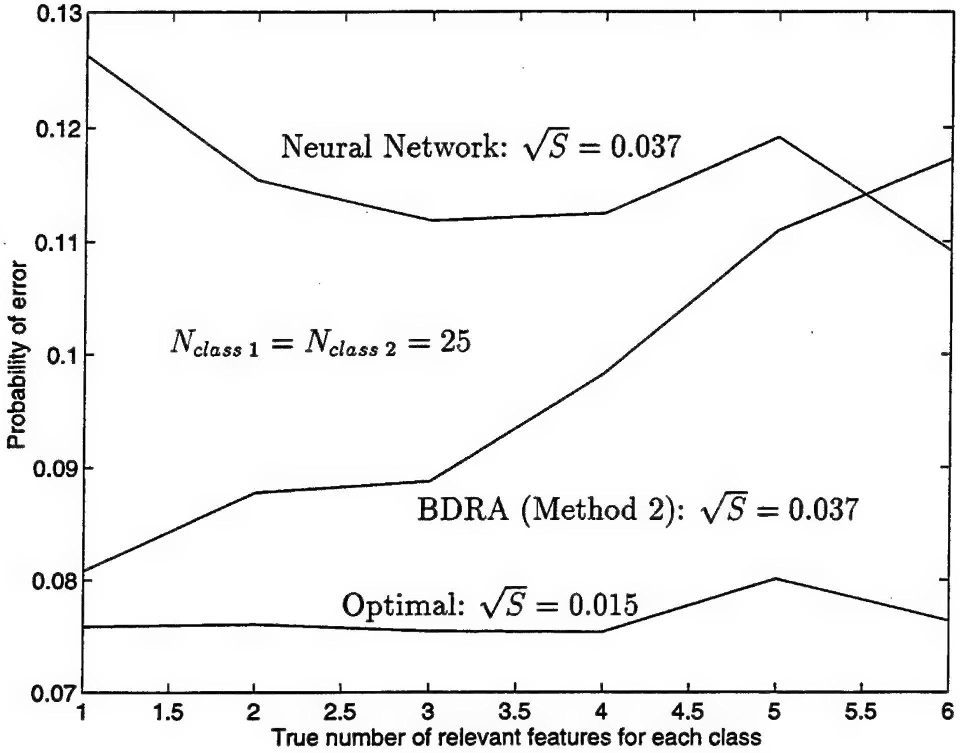


Figure 20: Performance comparison of Figure 19 repeated using the BDRA and Method 2.

In Figure 20 the same situation appears as that shown in Figure 19 except that now Method 2 is used for the BDRA instead of Method 1. It can be seen by comparing Figures 19 and 20 that both methods for the BDRA perform similarly at reducing the data when it contains missing features. Notice, this is important in Chapter 6 (see Section 6.4) where the BDRA is applied to a high dimensional data set because, as it turns out, in such a case Method 2 is less computationally intensive than Method 1.

With these results, the probability of missing features appearing in the training data was increased to 0.25, and in this situation the neural network showed more of a relative performance loss than the BDRA (particularly so with less than three relevant features per class). However, overall performance for both tests was similar to that shown in Figures 19 and 20. Additionally, it was found that both tests were equally effective at classifying test observations which contained missing features (see formula (38)).

5.4 Summary

The performance of the BDRA has been discussed at classifying six binary valued features given the training data of each class contains missing feature information. In adapting the BDRA for missing features two methods were developed. In the first method, each missing feature was assumed to be uniformly distributed over the cardinality of possible values it can take on. With the second method, each missing feature was assigned to an additional discrete level, which was obtained by increasing the number of discrete levels for that feature by one. Overall, it was found that both methods produced similar results. However, similar to that found in Chapter 3, the BDRA (using either method) was demonstrated to be superior to a neural network at reducing irrelevant features from the training data. In general, the missing feature information did not appear to have a large impact on degrading classification performance (for the BDRA or the neural network), even when the probability was moderately high for missing features to occur.

Chapter 6

Application of the BDRA to Miscellaneous Problems in Classification

6.1 Introduction

In this last chapter, the BDRA is applied to three interesting problems in classification. In the first problem, the BDRA is applied to reducing the dimensionality of a data set that contains class-specific features, and its performance is compared to the method developed in [2]. With this comparison, the BDRA is shown to be an effective means of selecting binary valued features, which have been made class-specific in an ad hoc manner. In fact, performance results reveal that when using a small number of training data relative to feature dimensionality (and when class-specific features exist for each class), the BDRA outperforms the class-specific classifier of [2].

In the second problem, the BDRA is applied to the fusion of features extracted from sonar echoes generated by independent continuous wave (CW) and frequency modulated (FM) waveforms. With this problem, a feature vector consisting of five features taken from CW and FM track pairs is used for detecting targets in various littoral environments. To illustrate performance, the BDRA is trained and tested on nearly five thousand samples of real sonar data consisting of these five dimensional feature vectors. Overall, it is shown that the BDRA improves target recognition performance, over that of other methods, by using three of the five features and quantizing them to binary values.

With the final problem, the BDRA is trained and tested on what is known in the literature as the Australian Credit Card Data (ACCD), [66]. Note, the ACCD is based on the actual credit history of 690 applicants, and it consists of fifteen features (both continuous and discrete valued), including missing features. In terms of performance, the BDRA is shown to be far superior to a neural network at classifying the test data.

6.2 The BDRA Applied to the Selection of Class-Specific Features

In [2], a novel approach to reducing the dimensionality of a feature set was developed by reformulating the optimum Bayesian classifier for C classes, and given by

$$\arg\max_{1\leq k\leq C}\left[f\left(\{\mathbf{z}_j\}_{j=1,\ldots,C}\left|H_k\right.\right)P\left(H_k\right)\right] \tag{39}$$

into an equivalent class-specific classifier, having the form

$$\arg\max_{1\leq k\leq C}\left[\frac{f\left(\mathbf{z}_k|H_k\right)}{f\left(\mathbf{z}_k|H_k,\theta_k=\theta_k^0\right)}P\left(H_k\right)\right] \tag{40}$$

where (see Chapters 2 and 3 for more on notation)

θ_k completely parameterizes the data, $\mathbf{x_k}$, representing the k^{th} class, H_k;

$\mathbf{z}_k = T_j\left(x\right)$ is a sufficient statistic for θ_k;

$f\left(\mathbf{z}_k|H_k,\theta_k=\theta_k^0\right)$ is a normalizing distribution and is the same for all k.

Notice that formula (39) can be expressed as formula (40) because each class has a unique sufficient statistic, z_k, which captures all relevant information about the parameter θ_k. In this way, formula (40) has important implications for data reduction. That is, the effects of the curse of dimensionality can be reduced if $f\left(\mathbf{z}_k|H_k\right)$ of formula (40) is estimated, as opposed to the higher dimensional $f\left(\left\{\mathbf{z}_j\right\}_{j=1,\ldots,C}|H_k\right)$ of formula (39). Given that, the problem addressed here is to determine if the BDRA can effectively reduce irrelevant information from a training data set which contains class-specific features. As a measure of performance, the probability of error for the BDRA is compared to the probability of error for formula (40).

To illustrate the class-specific classifier, consider Figure 21 below where the probability of error is plotted versus the true number of class-specific features (out of a total of six binary valued features) for each of two classes. In this figure, probability of error curves are shown for formula (16) of Chapter 3 (CBT), the class-specific version of formula (16) (CBT (class-specific)), and the optimal test.

In this case, optimal error probabilities have been constrained to be ≥ 0.05 and ≤ 0.1 (for more on this and the notation used here, see the text accompanying Figures 8 through 14 in Chapter 3).

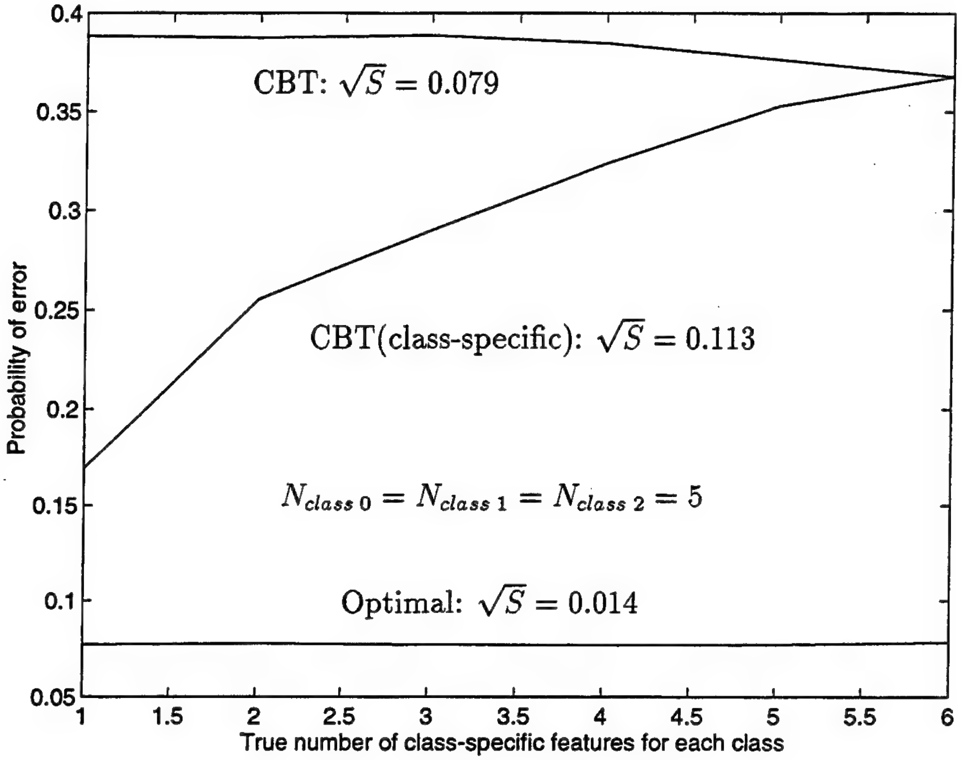


Figure 21: Performance of a class-specific classifier with binary valued features, and five samples of training data for each class.

Observe in Figure 21 that class 0 represents the common null class, and error probabilities in this figure (also in Figures 22 and 23) are based only on classifying classes 1 and 2. With this, in generating class-specific features for each class an ad hoc approach has been adopted using simulated data. That is, for a given true number of class-specific features those features which are class-specific, out of the total six, are determined randomly for each class. The remaining features (i.e., those which are not class-specific) are then distributed according to the null hypothesis, H_0. Additionally, bit probabilities for the class-specific features are determined using a Gaussian mixture distribution with a random number of modes (up to six modes), and bit probabilities for the commonly distributed features are based on a single Gaussian distribution. Also, results in Figure 21 are based on an average of two hundred fifty independent trials where, for each trial, ten thousand independent samples of test data were generated. Further, there are five samples of training data for each class (including the null class), and the sample standard deviations, $\sqrt{S}$, are given for each test.

It can bee seen in Figure 21 that the class-specific classifier based on formula (40) is superior to the classifier based on formula (39) by achieving an overall lower probability of error. However, the class-specific classifier also shows a slightly higher sample standard deviation in the probability of error than does the non-class-specific classifier. Notice, it can be seen that as more class-specific features are added to the feature vectors of each class the performance of both classifiers becomes similar. With this, and as expected, when all features are class-specific there is no difference in performance between the two methods.

An apparent observation in Figure 21 is that an insufficient number of training data causes the performance of both classifiers to be significantly above optimal. Given that, the effect of additional training samples is illustrated in Figure 22. In this case, the situation of Figure 21 is repeated except that each class now contains fifty samples of training data. As expected, observe in this figure that not only

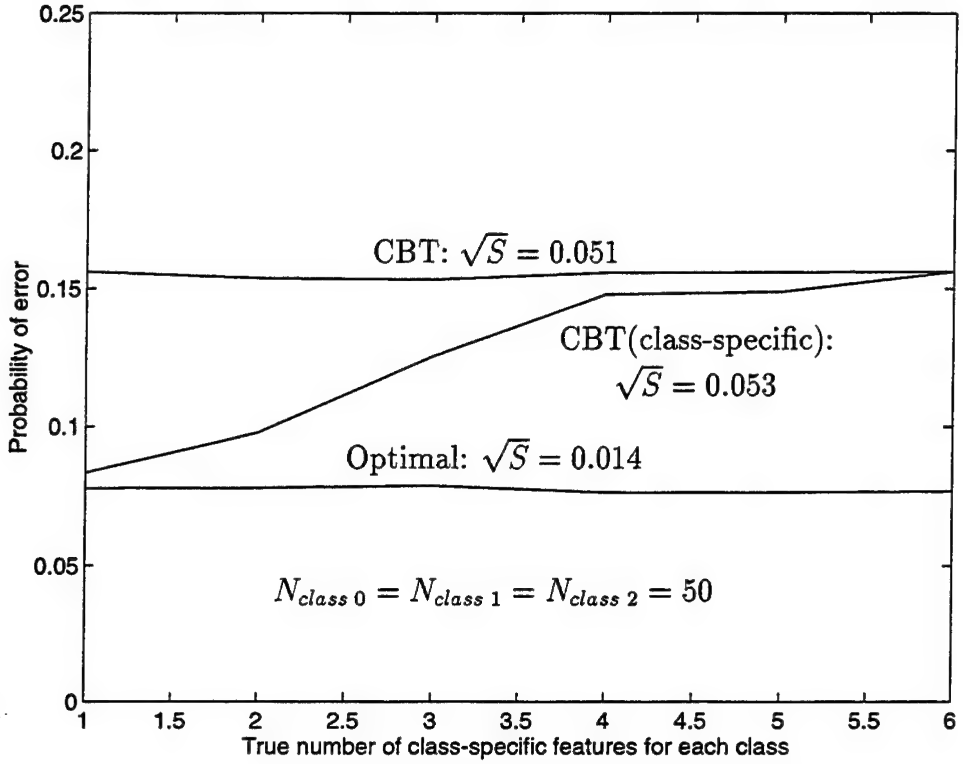


Figure 22: Performance of Figure 21 repeated with fifty samples of training data for each class.

has additional training data lowered the error probabilities of both classifiers (and with this the sample standard deviations are significantly less), but performance of the two methods is also closer.

In Figure 23 below, it is demonstrated that performance can be improved for the situation of Figure 21 if the BDRA is applied to the training data. Notice, in this figure that the error probability curves of Figure 21 are replotted, and results are shown for the BDRA applied using three different methods (results for each of these methods were produced independently using the same randomly generated symbol probabilities, but with different randomly generated data). In the first

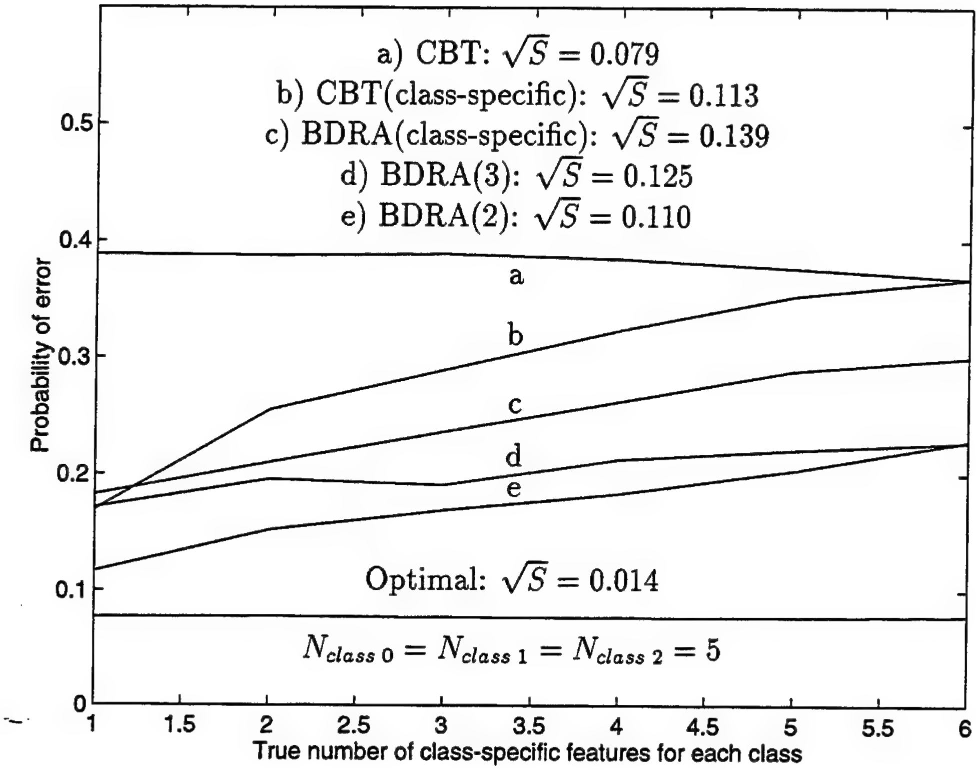


Figure 23: Performance of the BDRA with class-specific features when applied to the situation of Figure 21.

method, a class-specific version of the BDRA is used and labeled as BDRA(class-specific). Note, this method is essentially based on formula (40) in that feature reduction is performed separately on each of the two relevant classes versus the null, class 0. It can be seen that except for the case of one class-specific feature per class, this version of the BDRA improves performance. However, in the next method the BDRA is applied to all three classes simultaneously and it appears as BDRA(3). Clearly, results shown for BDRA(3) represent an overall improvement, but, the last method shown as BDRA(2) offers the best performance. In this method, the BDRA is applied simultaneously to only the two relevant classes (i.e., class 1 and class 2). Thus, feature reduction, and selection, using BDRA(2) is based only on those features which best discriminate class 1 from class 2, and without directly observing the null.

The results shown in Figure 23 reveal that the BDRA, and in particular one which is applied to only the relevant classes, is able to improve performance by effectively reducing the dimensionality of a training data set based on the empirical statistics of that data. Therefore, when automatically selecting relevant features, using a limited amount of training data, it is important to measure the impact that removal of irrelevant features has on overall discrimination capability amongst the relevant classes.

6.3 The BDRA Applied to the Fusion of Features From Independent Sonar Echoes

In this section, the BDRA is used for the fusion of features extracted from sonar echoes generated by independent continuous wave (CW) and frequency modulated (FM) waveforms. These sonar echoes were obtained in several different littoral environments, and their purpose is to track and detect various surface ships and submarines. The complete data set used here represents more than two thousand pings, which is a total time duration of approximately fifteen hours. Notice, correct target recognition with this data presents an interesting challenge because the sample size of the nontarget training data is more than five times larger than that of the target, and the CW waveform is a much better detector than the FM waveform.

From this data, five features were extracted and formed into the following feature vector,

{ Chi-square Statistic, CW Doppler, FM Doppler, CW KLLR, FM KLLR }

where

Chi-square statistic is a measure of track similarity, and it is obtained from the normalized (by the estimation errors) product of the difference between the individual CW and FM track state estimates.

CW Doppler appears in knots and is measured from the CW processor.

FM Doppler appears in knots and is estimated from range rate.

CW KLLR is the Kinematic log likelihood ratio detection statistic for CW that is based on track innovation.

FM KLLR is the Kinematic log likelihood ratio detection statistic for FM that is based on track innovation.

Note, the track state is a four dimensional vector of position and velocity estimates in both coordinates, and track estimation error is a 4×4 matrix. Also, for each waveform target track estimation is performed by an Interacting Multiple Model (IMM) Kalman Filter (see [4]).

Based on this feature vector, the data was partitioned into two classes; that is, a target class and a nontarget class (this latter class is made up of background disturbances such as shipping noise and clutter). The target class consists of CW and FM features in which at least one waveform has been verified to originate from a valid target, while the nontarget class is made up of only nontarget features from each waveform. Notice, in order to correctly label the data identification of true targets was performed by comparing estimated tracks to those of the Global Positioning Satellite (GPS). Therefore, any track not identified as a true target, by default, automatically was labeled a nontarget. With this, the total training set size was 5774 samples of which $N_{target} = 848$, and $N_{nontarget} = 4926$. Actually, the original data contains nearly one hundred thousand track pairs that can be considered of the nontarget category. However, a form of track pruning, or gating, was employed to substantially reduce this number by ordering all Chi-square statistics. That is, for each track only the smallest Chi-square statistic was accepted (the track it most closely associated with), and all other larger Chi-square statistics involving this track were rejected (all other tracks it might also have been associated with).

Before applying the BDRA to this data it was necessary to threshold each feature into an initial set of discrete levels. This thresholding was based on experience examining the data, and as a result, four thresholds were chosen for each feature. Thus, with four discrete levels per feature the initial quantization complexity of this data was $M = 1024$. Table 1 below lists these thresholds where

Table 1: Threshold Settings for Each Feature Before Applying the BDRA

Discrete level	Chi-square statistic	CW & FM Doppler	CW & FM KLLR
1	7.78	1	2.3
2	50	5	6.8
3	100	10	20
4	100K	70	30

at each discrete level the upper bound is shown, and the lower bound is defined in the next lower level.

After the BDRA was applied to the data the initial quantization complexity of $M = 1024$ was reduced to a final quantization complexity of $M = 8$. With this, the computed empirical probability of error (see formula (18), and Figure 8 of Chapter 3) was reduced from 0.325 to 0.117. In reducing this data, it was found that the BDRA completely removed the FM features. Additionally, it reduced the Chi-square statistic, CW Doppler, and CW KLLR to binary valued features keeping, respectively, the thresholds of 7.78, 1, and 2.3. Thus, for correct target recognition the BDRA prefers to rely mostly on CW, and it only uses FM when it associates with CW through the Chi-square statistic. As it turns out, this is consistent with the fact that FM is known to perform poorly in this data, and this is considered further in Figure 24 below.

Performance results of applying the BDRA for fusing CW and FM features are illustrated in Figure 24. In this figure, the total number of true detected targets versus the number of false detections per hour appears for the BDRA (note, results for the BDRA have been determined by testing on the training data),

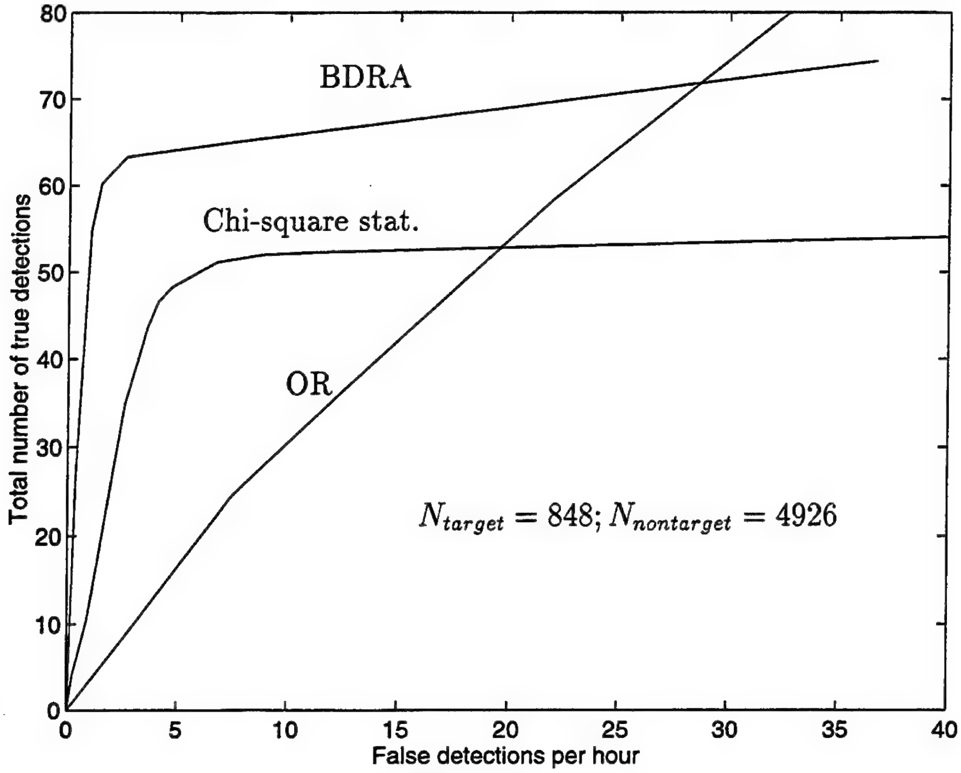


Figure 24: Target recognition performance comparison of the BDRA to the Chi-square statistic, and an OR detector.

the Chi-square statistic, and an OR detector (this detector is based on a logical OR of the individual CW and FM KLLR detector decisions). All results shown in Figure 24 represent detected target tracks, which have been converted from detected target pings using knowledge of the average number of pings contained in a track. It can be seen in Figure 24, that for low rates of false alert (the area of most interest) the BDRA is able to improve performance over the other methods. Notice, the OR detector performs poorly due to the high false alert rate of FM. Also, the Chi-square test is opportunity limited in that the target must exist in both waveforms in order for this test to detect it. On the other hand, the BDRA overcomes these limitations by selectively choosing those features associated with best performance.

6.4 The BDRA Applied to the Australian Credit Card Data

In the last problem addressed in this chapter the BDRA is trained and tested on the Australian Credit Card Data (ACCD). The ACCD is a data set that is often used by other authors for trying out their algorithms (for example, see [63, 66]), and it is based on the actual credit history of 690 applicants (307 applicants were issued credit, and 383 were denied credit). Also, the ACCD contains fifteen total features (six continuous and nine discrete), as well as some missing values (about five percent of the feature vectors have one or more missing values). Notice, that feature definitions are not supplied with this data in order to keep them confidential.

The ACCD is an interesting data set because it contains characteristics which make classification difficult. That is, because the ACCD contains a mix of fifteen discrete and continuous valued features, including missing features, a total of 690 samples (which must be partitioned into training and test data sets for both classes) is a relatively small data set for classifying accurately.

Because the ACCD contains missing features the methods of Chapter 5 are employed when applying the BDRA to the data. Specifically, the version of the BDRA called Method 2 in Chapter 5 is used as it was found to be significantly less computationally intensive with high dimensional data such as the ACCD. Recall,

Table 2: Initial Quantization for Each Feature of the ACCD

Feature	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Levels	3	3	2	5	4	15	10	2	2	2	2	2	3	3	2
Continuous(x)		x	x					x			x			x	x
Missing(x)	x	x		x	x	x	x							x	

in Method 2 each missing value of a feature is assigned to the same discrete level, which has been obtained by increasing the number of discrete levels for that feature by one.

To apply the BDRA to the ACCD it was necessary, as in the previous application, to discretize the continuous valued features. However, in this case a method of percentiles was chosen instead of using predefined values. For example, to obtain binary values for a feature the threshold was found (using both training and test data) which divided its sample size into two equal parts. As it turns out, binary values were chosen for each continuous valued feature because this produced the best classification performance for the ACCD. That is, the probability of error increased with the number of discrete levels for the continuous valued features. Further, if any of these continuous features also contained missing values, then its quantization was also increased to be ternary valued.

The initial quantization for each of the fifteen features of the ACCD is given in Table 2. Notice, also shown are labels for those features which were continuous, and those which have missing values. Thus, based on Table 2 the initial quantization complexity for this data is $M = 3.1104 \times 10^7$.

Table 3: Performance of the BDRA and a Neural Network With the ACCD

	BDRA	Neural Network
$P(e)$	0.135	0.284
$\sqrt{S}$	0.020	0.058

In applying the BDRA to the ACCD the experimental steps shown in [66] were followed. That is, the ACCD was randomly partitioned into training (518 samples) and test (172 samples) sets. Additionally, all results shown are based on an average of thirty independent trials. With this, the BDRA is compared to the the same neural network which was used in Chapter 5. However, the number of nodes in the input layer has obviously been increased to fifteen, and the number of nodes in the two successive hidden layers have respectively been increased to thirty and fifteen.

In Table 3, performance results are shown for the BDRA and a neural network (the neural network was initialized based on the minimum and maximum possible values of both the training and test data). It can be seen in this table that the BDRA reduces the probability of error by more than half as compared to the neural network, and it also shows less of a standard deviation in the error. With this, of the twenty three neural network algorithms tested in [66] using the ACCD the BDRA is in the top three (best performance is shown for an evolutionary type neural network with $P(e) = 0.115$). Also, the BDRA outperformed the results appearing in [63], which show $P(e) = 0.143$. However, in each of these cases the authors used only fourteen of the fifteen available features, thus, direct

Table 4: Final Quantization for Each Feature of the ACCD

Feature	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Levels	1	1	1	3.7	2.7	1.7	1.6	1	2	1	1.3	1	1	1	1

comparisons are more difficult to interpret. Further, the CGLRT of Appendix A was used in place of the CBT in the BDRA and it obtained a $P(e)$ of 0.157, with $\sqrt{S} = 0.024$.

With these results, Table 4 shows the final average quantization the BDRA produced over the thirty independent trials. As evidenced by the number of ones appearing in Table 4 the BDRA prefers to completely eliminate more than half of the available features. In fact, it obtains a final average quantization complexity of $M = 71.8$, which is a significant reduction of the data. Additionally, notice that the BDRA always keeps the ninth feature, and it only finds one of the continuous features (feature eleven) to be useful. This latter case helps to explain why the BDRA's performance diminishes as the quantization is increased for the continuous features.

Before concluding this section an important note is made about applying the BDRA to the ACCD, and this is relevant to any other potential applications involving training data sets which have a high dimensionality and a small sample size. In the initial application of the BDRA to the ACCD it was discovered that the empirical probability of error (see formula (18) of Chapter 3) was computed to be approximately 0.5, and this was subsequently reduced to around 0.34. In this case, the BDRA was performing poorly with the ACCD. However, this poor performance was only observed when the summation in formula (18) was taken over all possible test observations (i.e., $M = 3.1104 \times 10^7$ discrete symbols). Thus, performance was dramatically improved for the BDRA by summing formula (18) only over those test observations where a discrete symbol is represented in at least one of the training data sets of each class (i.e., for $N_{\mathbf{y}} = 1$ (extensions to $N_{\mathbf{y}} > 1$ are straightforward) summing only over those y_i where $x_{k,i} > 0$ or $x_{l,i} > 0$). This also required that formula (19) of Chapter 3 be redefined as $f\left(y_i = 1 | \mathbf{x}_k, H_k; x_{k,i} > 0 \text{ or } x_{l,i} > 0\right)$, which essentially amounts to renormalizing this formula so that it sums to unity over the range of i where, $x_{k,i} > 0$ or $x_{l,i} > 0$. Observe that with this modification a typical value of the empirical probability of error before data reduction was computed to be 0.33, and after applying the BDRA it was reduced to 0.14. In other words, even though the initial quantization complexity of the data is more than thirty one million discrete symbols, by employing the modification described above it is necessary for the BDRA to only consider those discrete symbols relevant to the training data. This helps to eliminate the curse of dimensionality, while lessening the number of computations required by the BDRA.

6.5 Summary

In this chapter, the BDRA was applied to several interesting problems in classification. With the first application, it was shown that when class-specific features are created for training data in an ad hoc manner (and with relatively small sample sizes), the BDRA can be used to reduce the data for improved classification performance. In fact, its performance was shown to be superior to the class-specific classifier. Also, in the second application the BDRA was used for fusing feature information from sonar echoes which were produced by independent CW and FM waveforms. In this case, the BDRA was shown to be more effective, at correct target recognition, than a Chi-square statistic test and an OR detector. The final application involved applying the BDRA to the Australian credit card data, and in this case the BDRA obtained a probability of error that was less than half that obtained by the same neural network used in previous chapters.

Appendix A

Results Using Empirically Generated Data

Simulated performance results are shown below for the CBT and the SGLRT of, respectively, formulas (4) and (10) of Chapter 2, and the CGLRT given by

$$\frac{f\left(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y} | \hat{\mathbf{p}}_k, \hat{\mathbf{p}}_l, H_k\right)}{f\left(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y} | \hat{\mathbf{p}}_k, \hat{\mathbf{p}}_l, H_l\right)} = \prod_{i=1}^{M} \frac{\hat{p}_{k,i,H_k}^{x_{k,i}+y_i} \hat{p}_{l,i,H_k}^{x_{l,i}}}{\hat{p}_{k,i,H_l}^{x_{k,i}} \hat{p}_{l,i,H_l}^{x_{l,i}+y_i}} \underset{H_l}{\overset{H_k}{\gtrless}} \tau \tag{41}$$

where symbol probabilities for the i^{th} are obtained from ML estimates (see, [12]) based on the training and test data, or,

$$\hat{p}_{k,i;H_k} = \frac{x_{k,i} + y_i}{N_k + N_{\mathbf{y}}}, \; \hat{p}_{l,i;H_l} = \frac{x_{l,i} + y_i}{N_l + N_{\mathbf{y}}},$$
$$\hat{p}_{k,i;H_l} = \frac{x_{k,i}}{N_k}, \text{ and } \hat{p}_{l,i;H_k} = \frac{x_{l,i}}{N_l}. \tag{42}$$

Results for the CBT, SGLRT, and the CGLRT are displayed using operating characteristic (OC) curves (i.e., the probability of correct recognition (Pd) versus the probability of false recognition (Pfa)), and the following items list specific information about the simulations:

- Each OC result is based on 10,000 independent iterations.
- The number of quantizing symbols is $M = 8$.
- At each iteration the symbol probabilities are generated according to a multivariate uniform distribution, and using these the training and test data are generated.
- To generate the training and test data the symbol probabilities above are modified in the Mth symbol probability.
- $p_{class\ 1,M(training)} = 0.005$ and $p_{class\ 1,M(test)} = 0.125$.

- The training data sizes are $N_{class\ 1} = 50$ and $N_{class\ 2} = 250$.
- Two test observation sizes, $N_{\mathbf{y}}$, of 2 and 25 are shown.
- The optimal curves in the OC are found by using the actual symbol probabilities of the test data.

Notice, the M^{th} symbol probability is modified to reflect a possible mismatch between the training data of class 1 and the test data, and indeed it is here that the main differences between the tests are to be found.

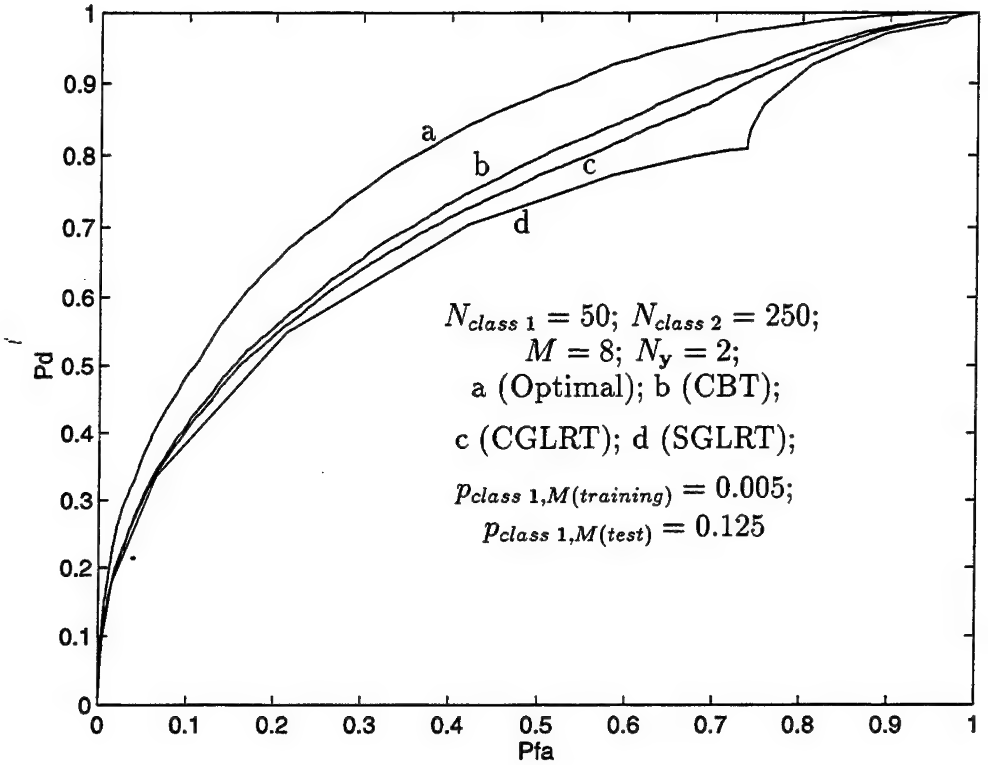


Figure 25: Simulated performance comparison of the CBT, CGLRT, and the SGLRT where $N_{\mathbf{y}} = 2$.

In Figure 25 (above) and Figure 26 (below) two OC plots appear for the case of $p_{class\ 1,M(training)} = 0.005$ and $p_{class\ 1,M(test)} = 0.125$. Also, Figure 25 represents the situation in which $N_{\mathbf{y}} = 2$, while that of Figure 26 represents $N_{\mathbf{y}} = 25$. Notice, in both of these figures a relatively severe mismatch between the testing and training distributions exists and, as a result, performance for all tests fall below optimal. However, because the CBT and the CGLRT use test observations to help infer the true symbol probabilities, both of these tests outperform the SGLRT. Observe, this is particularly true in Figure 26 ($N_{\mathbf{y}} = 25$), where the SGLRT's low estimate of $p_{class\ 1,M(training)}$ is causing it to make classification errors as the frequency of symbol M increases in the test data. Also, it can be seen that the SGLRT does not appear concave, and this is due to poor estimates of the symbol probabilities. Further, both of these figures demonstrate the similarity between the CBT and the CGLRT, and their performance improvement with test observations as predicted by Figure 4 of Chapter 2.

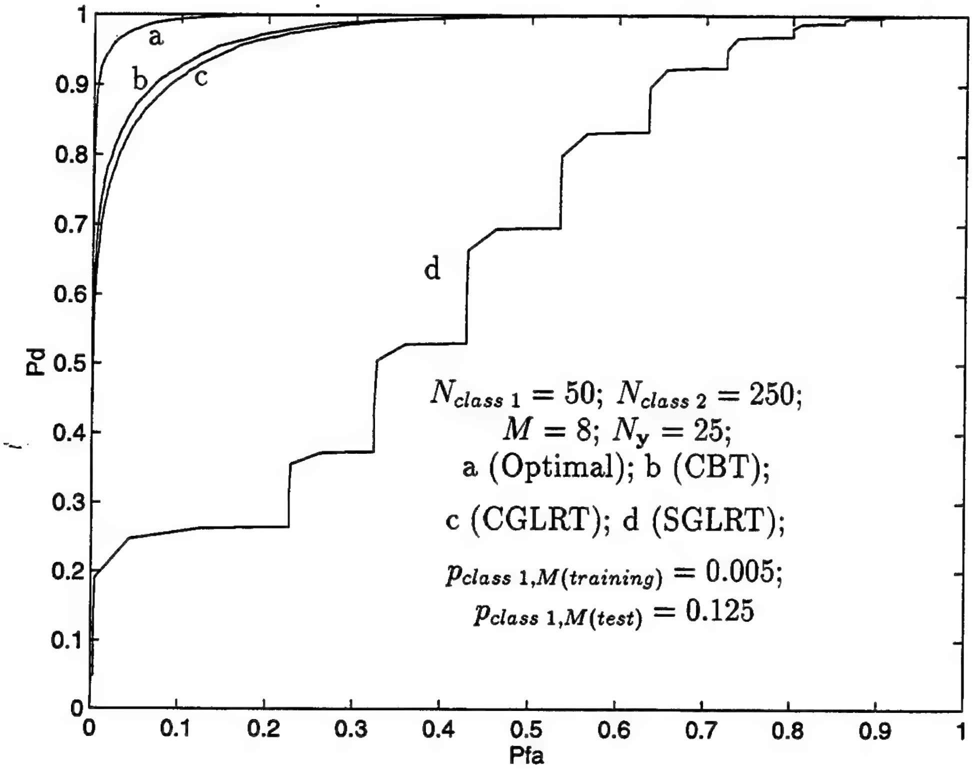


Figure 26: Simulated performance comparison of the CBT, CGLRT, and the SGLRT where $N_{\mathbf{y}} = 25$.

In general, the trend in performance appearing in Figures 25 and 26 continues as the difference between $p_{class\ 1,M(training)}$ and $p_{class\ 1,M(test)}$ increases. That is, performance of the SGLRT falls off more rapidly than either the CBT or the CGLRT, and this is because the combined tests are able to extract distributional information from the test observations. But, it is also noted that if the mismatched symbol appears substantially less often in the test data than predicted by the training data (e.g., $p_{class\ 1,M(training)} = 0.125$ and $p_{class\ 1,M(test)} = 0.005$), the performance loss in all three tests is not as severe.

In Figure 27 below empirical results are given for comparing performance of the CBT, formula (14) in Chapter 2, to that of the KST, formula (15) of the same chapter. In this figure, two curves appear for each test where $N_{class\ 1} =$

$N_{class\,2} = 16$, and $N_{class\,1} = N_{class\,2} = 50$. Also, there are a total of $M = 8$ discrete symbols, and the results are based on 10,000 independent trials where, at each trial, the true symbol probabilities are Dirichlet distributed. With this, the axis labels are the probability of declaring statistical similarity when true (Pd) versus the probability of declaring statistical similarity when false (Pfa). Clearly, it is apparent in this figure, and consistent with Figure 7 of Chapter 2, that the CBT overall performs better than the KST. Also, it can be seen that, as expected, the performance of both tests becomes similar as the sample size increases.

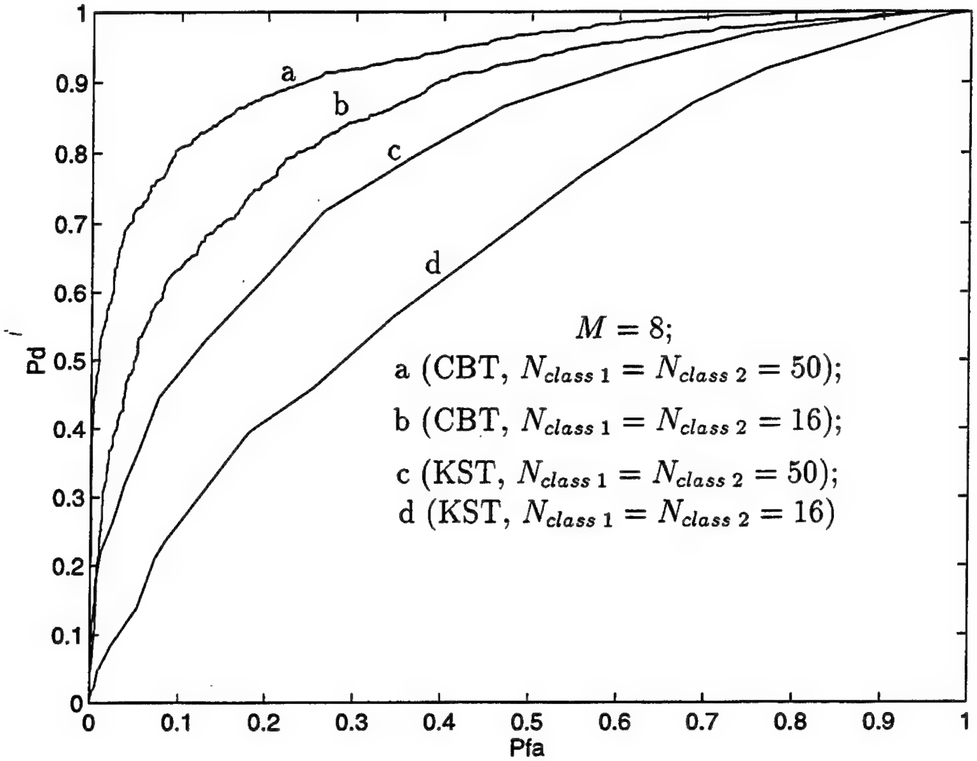


Figure 27: Simulated performance comparison of the CBT and the KST.

Appendix B

Development of the Combined Bayes Test

The CBT of formula (4), Chapter 2, is based on solving an integral expression of the type given by (this integral is solved for class k true, and for class l recall from formula (1) that k and l are exchangeable)

$$f\left(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y}|H_k\right) = \int_{\mathbf{p}_k} \int_{\mathbf{p}_l} f\left(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y}|\mathbf{p}_k, \mathbf{p}_l, H_k\right) f\left(\mathbf{p}_k\right) f\left(\mathbf{p}_l\right) d\mathbf{p}_k d\mathbf{p}_l, \qquad (43)$$

with each integration taken over the M dimensional unit simplex, [51].

Now, because of independence assumptions (see formula (1) of Chapter 2), formula (43) can be rewritten as

$$f\left(\mathbf{x}_k, \mathbf{x}_l, \mathbf{y}|H_k\right) = \int_{\mathbf{p}_k} f\left(\mathbf{x}_k, \mathbf{y}|\mathbf{p}_k, H_k\right) f\left(\vec{p}_k\right) d\mathbf{p}_k \int_{\mathbf{p}_l} f\left(\mathbf{x}_l|\mathbf{p}_l, H_k\right) f\left(\mathbf{p}_l\right) d\mathbf{p}_l \quad (44)$$

where (notice in formula (46) $\mathbf{x}_l$ is independent of H_k)

$$f\left(\mathbf{x}_k, \mathbf{y}|\mathbf{p}_k, H_k\right) = N_k! N_{\mathbf{y}}! \prod_{i=1}^{M} \frac{p_{k,i}^{x_{k,i}+y_i}}{x_{k,i}! y_i!} \qquad (45)$$

$$f\left(\mathbf{x}_l|\mathbf{p}_l, H_k\right) = f\left(\mathbf{x}_l|\mathbf{p}_l\right) = N_l! \prod_{i=1}^{M} \frac{p_{l,i}^{x_{l,i}}}{x_{l,i}!} \qquad (46)$$

and (note, for $f\left(\mathbf{p}_l\right)$ simply substitute l for k in $f\left(\mathbf{p}_k\right)$)

$$f\left(\mathbf{p}_k\right) = (M-1)! \mathcal{I}_{\left\{\sum_{i=1}^{M} p_{k,i}=1\right\}}. \qquad (47)$$

Observe that $f\left(\mathbf{p}_k\right)$ is uniformly distributed and represented by a Dirichlet distribution, [29, 33], which is also called the multivariate beta density, [45, 51]. In general, the form of this distribution is nonuniform (see formula (3) of Chapter 2) except, as in this case, when its parameters are selected to be unity, [8]. With this, the marginal probability formulas of the uniform Dirichlet are given by (these formulas are used below),

$$f\left(p_{k,i}|p_{k,i+1},p_{k,i+2},\ldots,p_{k,M}\right) = \begin{cases} (M-1)\left(1-p_{k,i}\right)^{M-2}\mathcal{I}_{\{(0,1)\}} & i=M \\ \frac{(i-1)}{\left(1-\sum_{j=i+1}^{M}p_{k,i}\right)}\left(1-\frac{p_{k,i}}{1-\sum_{j=i+1}^{M}p_{k,i}}\right)^{i-2}\mathcal{I}_{\left\{\left(0,1-\sum_{j=i+1}^{M}p_{k,i}\right)\right\}} & 1<i<M \\ \mathcal{I}_{\left\{\sum_{i=1}^{M}p_{k,i}=1\right\}} & i=1 \end{cases} \tag{48}$$

Now, the first integral in formula (44) is worked on next, and after factoring $f\left(\mathbf{p}_k\right)$ using formula (48) it is given by

$$\begin{aligned} f\left(\mathbf{x}_k,\mathbf{y}|H_k\right) &= \int_{\mathbf{p}_k} f\left(\mathbf{x}_k,\mathbf{y}|\mathbf{p}_k,H_k\right) f\left(p_{k,1}|p_{k,2},\ldots,p_{k,M}\right) \\ &\quad \times f\left(p_{k,2}|p_{k,3},\ldots,p_{k,M}\right)\ldots f\left(p_{k,M}\right)d\mathbf{p}_k. \end{aligned} \tag{49}$$

Continuing, formulas (45) and (48) are used in the first two distributions after the integral sign of formula (49), and then the integral over $\mathbf{p}_k$ is broken down into M separate integrals producing

$$\begin{aligned} f\left(\mathbf{x}_k,\mathbf{y}|H_k\right) &= \int_0^1\ldots\int_0^{1-\sum_{i=2}^{M}p_{k,i}} \frac{N_k!N_{\mathbf{y}}!}{\left[\prod_{i=1}^{M}x_{k,i}!y_i!\right]} \\ &\quad \times \left[\prod_{i=1}^{M}p_{k,i}^{x_{k,i}+y_i}\right]\left[\mathcal{I}_{\left\{\sum_{i=1}^{M}p_{k,i}=1\right\}}\right] \\ &\quad \times f\left(p_{k,2}|p_{k,3},\ldots,p_{k,M}\right)\ldots f\left(p_{k,M}\right)dp_{k,1}\ldots dp_{k,M}. \end{aligned} \tag{50}$$

This is integrated with respect to $p_{k,1}$ yielding

$$\begin{aligned} &= \int_0^1\ldots\int_0^{1-\sum_{i=3}^{M}p_{k,i}} \frac{N_k!N_{\mathbf{y}}!}{\left[\prod_{i=1}^{M}x_{k,i}!y_i!\right]} \\ &\quad \times \left[\prod_{i=2}^{M}p_{k,i}^{x_{k,i}+y_i}\right]\left[\left(1-\sum_{i=2}^{M}p_{k,i}\right)^{x_{k,1}+y_1}\right] \\ &\quad \times f\left(p_{k,2}|p_{k,3},\ldots,p_{k,M}\right)\ldots f\left(p_{k,M}\right)dp_{k,2}\ldots dp_{k,M}. \end{aligned} \tag{51}$$

From here, formula (48) is used in $f\left(p_{k,2}|p_{k,3},\ldots,p_{k,M}\right)$ of formula (51) so that

$$
\begin{aligned}
f\left(\mathbf{x}_k, \mathbf{y}|H_k\right) &= \int_0^1 \cdots \int_0^{1-\sum_{i=3}^{M} p_{k,i}} \frac{N_k! N_{\mathbf{y}}!}{\left[\prod_{i=1}^{M} x_{k,i}! y_i!\right]} \\
&\quad \times \left[\prod_{i=2}^{M} p_{k,i}^{x_{k,i}+y_i}\right] \left[\left(1-\sum_{i=2}^{M} p_{k,i}\right)^{x_{k,1}+y_1}\right] \left[\frac{1}{\left(1-\sum_{i=3}^{M} p_{k,i}\right)}\right] \\
&\quad \times f\left(p_{k,3}|p_{k,4}, \ldots, p_{k,M}\right) \ldots f\left(p_{k,M}\right) dp_{k,2} \ldots dp_{k,M}.
\end{aligned} \tag{52}
$$

Before integrating formula (52) with respect to $p_{k,2}$ an integral expression is required that is given by

$$
\int_0^A (A-w)^a w^b dw = A^{a+b+1} \frac{a!b!}{(a+b+1)!} \tag{53}
$$

and using this to do the integration results in

$$
\begin{aligned}
f\left(\mathbf{x}_k, \mathbf{y}|H_k\right) &= \int_0^1 \cdots \int_0^{1-\sum_{i=4}^{M} p_{k,i}} \frac{N_k! N_{\mathbf{y}}!}{\left[\prod_{i=1}^{M} x_{k,i}! y_i!\right]} \\
&\quad \times \left[\prod_{i=3}^{M} p_{k,i}^{x_{k,i}+y_i}\right] \left[\left(1-\sum_{i=3}^{M} p_{k,i}\right)^{x_{k,1}+y_1+x_{k,2}+y_2}\right] \\
&\quad \times \frac{\left(x_{k,1}+y_1\right)! \left(x_{k,2}+y_2\right)!}{\left(x_{k,1}+y_1+x_{k,2}+y_2+1\right)!} \\
&\quad \times f\left(p_{k,3}|p_{k,4}, \ldots, p_{k,M}\right) \ldots f\left(p_{k,M}\right) dp_{k,3} \ldots dp_{k,M}.
\end{aligned} \tag{54}
$$

Now, the procedure employed in formulas (51) through (54) is repeated with respect to $p_{k,3}$ producing

$$
\begin{aligned}
f\left(\mathbf{x}_k, \mathbf{y}|H_k\right) &= \int_0^1 \cdots \int_0^{1-\sum_{i=5}^{M} p_{k,i}} \frac{2N_k! N_{\mathbf{y}}!}{\left[\prod_{i=1}^{M} x_{k,i}! y_i!\right]} \\
&\quad \times \left[\prod_{i=4}^{M} p_{k,i}^{x_{k,i}+y_i}\right] \left[\left(1-\sum_{i=4}^{M} p_{k,i}\right)^{x_{k,1}+y_1+x_{k,2}+y_2+x_{k,3}+y_3}\right] \\
&\quad \times \frac{\left(x_{k,1}+y_1\right)! \left(x_{k,2}+y_2\right)! \left(x_{k,3}+y_3\right)!}{\left(x_{k,1}+y_1+x_{k,2}+y_2+x_{k,3}+y_3+2\right)!} \\
&\quad \times f\left(p_{k,4}|p_{k,5}, \ldots, p_{k,M}\right) \ldots f\left(p_{k,M}\right) dp_{k,4} \ldots dp_{k,M}
\end{aligned} \tag{55}
$$

and likewise using the same procedure with respect to $p_{k,4}$ yields

$$
= \int_0^1 \cdots \int_0^{1-\sum_{i=6}^{M} p_{k,i}} \frac{3 \times 2N_k! N_{\mathbf{y}}!}{\left[\prod_{i=1}^{M} x_{k,i}! y_i!\right]}
$$

$$
\begin{aligned}
&\times\left[\prod_{i=5}^{M} p_{k,i}^{x_{k,i}+y_i}\right]\left[\left(1-\sum_{i=5}^{M} p_{k,i}\right)^{x_{k,1}+y_1+x_{k,2}+y_2+x_{k,3}+y_3+x_{k,4}+y_4}\right] \\
&\times\frac{(x_{k,1}+y_1)!\,(x_{k,2}+y_2)!\,(x_{k,3}+y_3)!\,(x_{k,4}+y_4)!}{(x_{k,1}+y_1+x_{k,2}+y_2+x_{k,3}+y_3+x_{k,4}+y_4+3)!} \\
&\times f\left(p_{k,5}|p_{k,6},\ldots,p_{k,M}\right)\ldots f\left(p_{k,M}\right)dp_{k,5}\ldots dp_{k,M}. \qquad (56)
\end{aligned}
$$

Continuing in this way a final result is obtained, which is given by

$$
\begin{aligned}
f\left(\mathbf{x}_k,\mathbf{y}|H_k\right) &= \frac{(M-1)!N_k!N_{\mathbf{y}}!}{\left[\prod_{i=1}^{M} x_{k,i}!y_i!\right]}\frac{\left[\prod_{i=1}^{M}\left(x_{k,i}+y_i\right)!\right]}{\left[\left(\sum_{i=1}^{M}\left(x_{k,i}+y_i\right)+M-1\right)!\right]} \\
&= \frac{(M-1)!N_k!N_{\mathbf{y}}!}{(N_k+N_{\mathbf{y}}+M-1)!}\prod_{i=1}^{M}\frac{(x_{k,i}+y_i)!}{x_{k,i}!y_i!} \qquad (57)
\end{aligned}
$$

Returning now to formula (44), the second integral expression is evaluated by integrating as before, and this results in (where, $f\left(\mathbf{x}_l|H_k\right) = f\left(\mathbf{x}_l\right)$, see formula (46))

$$
\begin{aligned}
f\left(\mathbf{x}_l\right) &= \frac{(M-1)!N_l!}{\left[\prod_{i=1}^{M} x_{l,i}!\right]}\frac{\left[\prod_{i=1}^{M} x_{l,i}!\right]}{\left[\left(\sum_{i=1}^{M}\left(x_{l,i}\right)+M-1\right)!\right]} \\
&= \frac{(M-1)!N_l!}{(N_l+M-1)!} \qquad (58)
\end{aligned}
$$

A few notes about the formulas in (57) and (58) are necessary. First, the result in formula (58) can be found by simply eliminating any test observations in formula (57). Also, notice that without test observations $f\left(\mathbf{x}_l\right)$ in formula (58) is uniformly distributed. In other words, the likelihood of all training data being the same symbol is equal to them being different symbols. However, with test observations, the same procedure produces a nonuniform result for $f\left(\mathbf{x}_k,\mathbf{y}|H_k\right)$ as shown in formula (57). That is, formula (57) is distributed according to the number of occurrences of training and test data.

Using formulas (57) and (58) the integral expression in formula (44) can then be solved, or

$$
f\left(\mathbf{x}_k,\mathbf{x}_l,\mathbf{y}|H_k\right) = \frac{\left[(M-1)!\right]^2 N_k!N_l!N_{\mathbf{y}}!}{(N_k+N_{\mathbf{y}}+M-1)!(N_l+M-1)!}\prod_{i=1}^{M}\frac{(x_{k,i}+y_i)!}{x_{k,i}!y_i!} \qquad (59)
$$

where the equivalent formula for class l (i.e., H_l) is obtained by a substitution of l for k. Finally, the combined Bayes test of formula (4) in Chapter 2 then becomes the ratio of formula (59) to its equivalent formula under class l.

Appendix C

Development of $f\left(\mathbf{y} \mid \mathbf{x}_k, H_k\right)$ from Probabilistic Considerations

In Chapter 3, it was previously mentioned that the distribution $f\left(\mathbf{y}|\mathbf{x}_k, H_k\right)$ shown as formula (17) was required for the error probability formula of (18). Here, this distribution is developed form probabilistic considerations, and to do this, a random vector is first defined which is the sum of the unknown test vector and the k^{th} training data set, or, $\mathbf{s} = \mathbf{y} + \mathbf{x}_k$.

Now, under a Dirichlet prior for the symbol probabilities given by

$$f\left(\mathbf{p}_k\right) = (M-1)!\mathcal{I}_{\left\{\sum_{i=1}^{M} p_{k,i}=1\right\}} \tag{60}$$

the distribution of $\mathbf{s}$ can be determined. However, before applying the Dirichlet, the distribution of $\mathbf{s}$ conditioned on $\mathbf{p_k}$ and H_k is multinomial and it appears as

$$f\left(\mathbf{s}|\mathbf{p}_k, H_k\right) = \left(N_k + N_{\mathbf{y}}\right)! \prod_{i=1}^{M} \frac{p_{k,i}^{s_i}}{s_i!}. \tag{61}$$

Then, after applying the Dirichlet the distribution of $\mathbf{s}$ is given by

$$f\left(\mathbf{s} \mid H_k\right) = \frac{1}{\begin{pmatrix} N_k + N_{\mathbf{y}} + M - 1 \\ N_k + N_{\mathbf{y}} \end{pmatrix}}. \tag{62}$$

Notice, as in formula (8) of Chapter 2, the notation $\begin{pmatrix} N_k + N_{\mathbf{y}} + M - 1 \\ N_k + N_{\mathbf{y}} \end{pmatrix}$ means the number of ways that $N_k + N_{\mathbf{y}}$ samples can be arranged amongst $M-1$ discrete symbols. Also, the result in formula (62) means that $\mathbf{s}$ is, with equal probability, any valid value.

Continuing, conditioned on $\mathbf{s}$, any way of choosing the $N_{\mathbf{y}}$ test observations is also equiprobable, and this has the distribution

$$f\left(\mathbf{y} \mid \mathbf{s}, H_k\right) = \frac{\prod_{i=1}^{M} \begin{pmatrix} s_i \\ y_i \end{pmatrix}}{\begin{pmatrix} N_k + N_{\mathbf{y}} \\ N_{\mathbf{y}} \end{pmatrix}}. \tag{63}$$

It is also apparent from the definition of $\mathbf{s}$ that

$$f\left(\mathbf{x}_k \mid \mathbf{y}, \mathbf{s}, H_k\right) = \mathcal{I}_{\{\mathbf{s}=\mathbf{y}+\mathbf{x}_k\}}. \tag{64}$$

The distributions of formulas (62), (63), and (64) result in

$$f\left(\mathbf{x}_k, \mathbf{y}, \mathbf{s} \mid H_k\right) = \frac{\mathcal{I}_{\{\mathbf{s}=\mathbf{y}+\mathbf{x}_k\}} \prod_{i=1}^{M} \begin{pmatrix} s_i \\ y_i \end{pmatrix}}{\begin{pmatrix} N_k + N_{\mathbf{y}} \\ N_{\mathbf{y}} \end{pmatrix} \begin{pmatrix} N_k + N_{\mathbf{y}} + M - 1 \\ N_k + N_{\mathbf{y}} \end{pmatrix}}. \tag{65}$$

From which we obtain

$$f\left(\mathbf{x}_k, \mathbf{y} \mid H_k\right) = \frac{\prod_{i=1}^{M} \begin{pmatrix} x_{k,i} + y_i \\ y_i \end{pmatrix}}{\begin{pmatrix} N_k + N_{\mathbf{y}} \\ N_{\mathbf{y}} \end{pmatrix} \begin{pmatrix} N_k + N_{\mathbf{y}} + M - 1 \\ N_k + N_{\mathbf{y}} \end{pmatrix}}. \tag{66}$$

Now, noting that $f\left(\mathbf{x}_k\right) = \begin{pmatrix} N_k + M - 1 \\ N_k \end{pmatrix}^{-1}$ (see formula (58) of Appendix B), the desired distribution is found, or,

$$\begin{aligned} f\left(\mathbf{y} \mid \mathbf{x}_k, H_k\right) &= \frac{f\left(\mathbf{x}_k, \mathbf{y} \mid H_k\right)}{f\left(\mathbf{x}_k\right)} \\ &= \frac{\prod_{i=1}^{M} \begin{pmatrix} x_{k,i} + y_i \\ y_i \end{pmatrix} \begin{pmatrix} N_k + M - 1 \\ N_k \end{pmatrix}}{\begin{pmatrix} N_k + N_{\mathbf{y}} \\ N_{\mathbf{y}} \end{pmatrix} \begin{pmatrix} N_k + N_{\mathbf{y}} + M - 1 \\ N_k + N_{\mathbf{y}} \end{pmatrix}} \\ &= \frac{\prod_{i=1}^{M} \begin{pmatrix} x_{k,i} + y_i \\ y_i \end{pmatrix}}{\begin{pmatrix} N_k + N_{\mathbf{y}} + M - 1 \\ N_{\mathbf{y}} \end{pmatrix}} \\ &= \frac{\left(N_k + M - 1\right)! \left(N_{\mathbf{y}}\right)!}{\left(N_k + N_{\mathbf{y}} + M - 1\right)!} \prod_{i=1}^{M} \frac{\left(x_{k,i} + y_i\right)!}{\left(x_{k,i}\right)! \left(y_i\right)!} \end{aligned} \tag{67}$$

which is the same as formula (17) of Chapter 3.

Appendix D

Mean and Variance of the Probability of Error for Dirichlet Distributed Symbol Probabilities

Under the assumption of an optimal test, and that there are two classes labeled k and l, the test chooses k if $y = i$ and $p_{k,i} \geq p_{l,i}$. Thus, for the probability of error we have (see, [34])

$$\begin{aligned} P(e|k) &= \sum_{i=1}^{M} P(e, y = i|k) & (68)\\ &= MP(e, y = 1|k) & (69)\\ &= MPr(p_{l,1} > p_{k,1}, y = 1 \mid k) & (70) \end{aligned}$$

where formulas (68) and (69) result from total probability and symmetry in the probabilities, and formula (70) is based on the definition of an error under k. But, formula (70) is also equivalent to

$$\begin{aligned} &= M \int_0^1 Pr(p_{l,1} > p_{k,1}, y = 1 \mid p_{k,1}, k)\, f(p_{k,1})\, dp_{k,1} & (71)\\ &= M \int_0^1 Pr(p_{l,1} > p_{k,1} \mid y = 1, p_{k,1}, k)\, Pr(y = 1 \mid p_{k,1}, k)\, f(p_{k,1})\, dp_{k,1} & (72) \end{aligned}$$

where formulas (71) and (72) result from another application of total probability, and conditional independence.

Now, using the definitions (see formula (48) of Appendix B),

$$\begin{aligned} f(p_{k,1}) &= (M-1)(1-p_{k,1})^{M-2} & (73)\\ Pr(y = 1 \mid p_{k,1}, k) &= p_{k,1} & (74)\\ Pr(p_{l,1} > p_{k,1}, y = 1 \mid p_{k,1}, k) &= \int_{p_{k,1}}^{1} (M-1)(1-p_{k,1})^{M-2} \\ &= (1-p_{k,1})^{M-1} & (75) \end{aligned}$$

formula (72) then becomes

$$
\begin{aligned}
&= M\int_0^1 \left(1-p_{k,1}\right)^{M-1} p_{k,1}\left(M-1\right)\left(1-p_{k,1}\right)^{M-2} dp_{k,1} && (76)\\
&= M\left(M-1\right)\int_0^1 \left(1-p_{k,1}\right)^{2M-3} p_{k,1} dp_{k,1} && (77)\\
&= \frac{1}{2}\left(\frac{M}{2M-1}\right). && (78)
\end{aligned}
$$

Note, formula (52) of Appendix B was used in obtaining formula (78), and it is clear in this result that under a uniform Dirichlet distribution, as M approaches infinity, the quantity $P\left(e|k\right)$ approaches 1/4.

Using the results above, the variance of $P\left(e|k\right)$ can be determined by first finding

$$
\begin{aligned}
\mathcal{E}\left\{P\left(e|k\right)^2\right\} &= M\int_0^1 \left(1-p_{k,1}\right)^{2M-2} p_{k,1}^2\left(M-1\right)\left(1-p_{k,1}\right)^{M-2} dp_{k,1}\\
&\quad +M(M-1)\int_0^1 \left(1-p_{j,1}\right)^{M-1} p_{j,1}\left(M-1\right)\left(1-p_{j,1}\right)^{M-2} dp_{j,1}\\
&\quad \times\int_0^1 \left(1-p_{k,1}\right)^{M-1} p_{k,1}\left(M-1\right)\left(1-p_{k,1}\right)^{M-2} dp_{k,1} && (79)\\
&= \frac{2M}{3\left(3M-1\right)\left(3M-2\right)}+\frac{M\left(M-1\right)}{4\left(2M-1\right)^2} && (80)
\end{aligned}
$$

and after subtracting from formula (80) the formula of (78) squared, produces the result

$$
\text{Var } P\left(e \mid k\right) = \frac{2}{3}\left(\frac{M}{\left(3M-1\right)\left(3M-2\right)}\right)-\frac{1}{4}\left(\frac{M}{\left(2M-1\right)^2}\right). \qquad (81)
$$

With this, it can be seen that as M approaches infinity the variance of $P\left(e \mid k\right)$ approaches a limit of zero.

The result in formula (78) can be straightforwardly extended to three classes (i.e., $C=3$) be redefining formula (70) as

$$
P\left(e|k;C=3\right) = M\left(1-Pr\left(p_{j,1}<p_{k,1},p_{l,1}<p_{k,1},y=1 \mid k\right)\right) \qquad (82)
$$

and using formulas (73), (74), and

$$
\begin{aligned}
Pr\left(p_{l,1}<p_{k,1},y=1 \mid p_{k,1},k\right) &= \int_0^{p_{k,1}} \left(M-1\right)\left(1-p_{k,1}\right)^{M-2}\\
&= 1-\left(1-p_{k,1}\right)^{M-1} && (83)
\end{aligned}
$$

formula (82) becomes

$$= \quad M\int_0^1 \left(1-(1-p_{k,1})^{M-1}\right)^2 p_{k,1}\,(M-1)\,(1-p_{k,1})^{M-2}\,dp_{k,1} \qquad (84)$$

$$= \quad \frac{M}{(2M-1)} - \frac{1}{3}\left(\frac{M}{3M-2}\right). \qquad (85)$$

Observe, that in the limit as M approaches infinity $P\,(e|k; C=3)$ approaches 7/18. Thus, as compared to formula (78), the limit of the average probability of error increases with the number of classes under Dirichlet distributed symbol probabilities.

Bibliography

[1] K. Abend and T. J. Harley, Jr., "Comments on "The Mean Accuracy of Statistical Pattern Recognizers'," *IEEE Transactions on Information Theory*, vol. 15, May 1969, pp. 420-421.

[2] P. M. Baggenstoss, "Class-Specific Feature Sets in Classification," *To appear in a future issue of the IEEE Transactions on Signal Processing.*

[3] B. Baygün and A. O. Hero III, "Optimal Simultaneous Detection and Estimation Under a False Alarm Constraint," *IEEE Transactions on Information Theory*, vol. 41, no. 3, May 1995, pp. 688-703.

[4] Y. Bar-Shalom and X. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, Course Notes, University of Connecticut, 1995.

[5] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory*, Wiley, New York, 1994.

[6] T. G. Birdsall and J. O. Gobien, "Sufficient Statistics and Reproducing Densities in Simultaneous Sequential Detection and Estimation," *IEEE Transactions on Information Theory*, vol. 19, no. 6, November 1973, pp. 760-768.

[7] C. M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.

[8] C. G. E. Boender and A. H. G. Rinnooy Kan, "A Multinomial Bayesian Approach to the Estimation of Population and Vocabulary Size," *Biometrika*, vol. 74, no. 4, 1987, pp. 849-856.

[9] L. J. Buturović, "Toward Bayes-Optimal Linear Dimension Reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 4, April 1994, pp. 420-423.

[10] L. L. Campbell, "Averaging Entropy," *IEEE Transactions on Information Theory*, vol. 41, no. 1, January 1995, pp. 338-339.

[11] B. P. Carlin and T. A. Louis, *Bayes and Empirical Bayes Methods for Data Analysis*, Chapman & Hall, London, 1996.

[12] G. Casella and R. L. Berger, *Statistical Inference*, Duxbury Press, Belmont, California, 1990.

[13] B. Chandrasekaran, "Independence of Measurements and the Mean Recognition Accuracy," *IEEE Transactions on Information Theory*, vol. 17, July 1971, pp. 452-456.

[14] B. Chandrasekaran and T. J. Harley, Jr., "Comments on "The Mean Accuracy of Statistical Pattern Recognizers'," *IEEE Transactions on Information Theory*, vol. 15, May 1969, pp. 421-423.

[15] M. H. DeGroot, *Probability and Statistics*, Addison-Wesley Publishing Company, Reading, Massachusetts, August 1987.

[16] M. Delampady and J. O. Berger, "Lower Bounds on Bayes Factors for Multinomial Distributions, With Applications to Chi-Square Tests of Fit," *The Annals of Statistics*, vol. 18, no. 3, 1990, pp. 1295-1316.

[17] H. Demuth and M. Beale, *Neural Network Toolbox*, The Math Works, Inc., Natick, MA, 1994.

[18] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer-Verlag, New York, NY, 1996.

[19] P. Dianconis and D. Freedman, "On the Uniform Consistency of Bayes Estimates for Multinomial Probabilities," *The Annals of Statistics*, vol. 18, no. 3, 1990, pp. 1317-1327.

[20] R. D. Dony and S. Haykin, "Neural Network Approaches to Image Compression," *Proceedings of the IEEE*, vol. 83, no. 2, February 1995, pp. 288-303.

[21] R. P. W. Duin, "The Mean Recognition Performance for Independent Distributions," *IEEE Transactions on Information Theory*, vol. 24, no. 3, April 1978, pp. 394-395.

[22] R. P. W. Duin, C. E. van Haersma Buma, and L. Roosma, "On the Evaluation of Independent Binary Features," *IEEE Transactions on Information Theory*, vol. 24, no. 2, March 1978, pp. 248-249.

[23] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, NY, 1973.

[24] Y. Ephraim, "Statistical-Model-Based Speech Enhancement Systems," *Proceedings of the IEEE*, vol. 80, no. 10, October 1992, pp. 1526-1555.

[25] K. Fukunaga, *Statistical Pattern Recognition*, Academic Press, Inc., Boston, 1990.

[26] K. Fukunaga and R. R. Hayes, "Effects of Sample Size in Classifier Design," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 8, August 1989, pp. 873-885.

[27] K. Fukunaga and D. Kessell, "Nonparametric Bayes Error Estimation Using Unclassified Samples," *IEEE Transactions on Information Theory*, vol. 19, 1973, pp. 434-440.

[28] I. J. Good, "The Bayes Factor Against Equiprobability of a Multinomial Population Assuming a Symmetric Dirichlet Prior," *The Annals of Statistics*, vol. 2, no. 5, 1974, pp. 977-987.

[29] I. J. Good and J. F. Crook, "The Robustness and Sensitivity of the Mixed-Dirichlet Bayesian Test for "Independence" in Contingency Tables," *The Annals of Statistics*, vol. 15, no. 2, 1987, pp. 670-693.

[30] M. Gutman, "Asymptotically Optimal Classification for Multiple Tests with Empirically Observed Statistics," *IEEE Transactions on Information Theory*, vol. 35, no. 2, March 1989, pp. 401-407.

[31] R. Hanson, J. Stutz, and P. Cheeseman, "Bayesian Classification Theory," *NASA Ames Research Center Technical Report*, no. FIA-90-12-7-01, December 1990.

[32] J. P. Hoffbeck and D. A. Landgrebe, "Covariance Matrix Estimation and Classification With Limited Training Data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, July 1996, pp. 763-767.

[33] R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics*, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1995.

[34] G. F. Hughes, "On the Mean Accuracy of Statistical Pattern Recognizers," *IEEE Transactions on Information Theory*, vol. 14, no. 1, January 1968, pp. 55-63.

[35] Q. Huo, H. Jiang, and C. Lee, "A Bayesian Predictive Classification Approach to Robust Speech Recognition," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1997, pp. 1547-1550.

[36] A. G. Jaffer and S. C. Gupta, "Coupled Detection-Estimation of Gaussian Processes in Gaussian Noise," *IEEE Transactions on Information Theory*, vol. 18, no. 1, January 1972, pp. 106-110.

[37] A. Jain and D. Zongker, "Feature Selection: Evaluation, Application, and Small Sample Size Performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, February 1997, pp. 153-158.

[38] R. E. Kass and A. E. Raftery, "Bayes Factors," *Journal of the American Statistical Association*, vol. 90, no. 430, June 1995, pp. 773-795.

[39] D. Kazakos, "Quantization Complexity and Training Sample Size in Detection," *IEEE Transactions on Information Theory*, vol. 24, no. 2, March 1978, pp. 229-237.

[40] J. Kittler, "Feature Set Search Algorithms," *Pattern Recognition and Signal Processing*, C. H. Chen, ed., Sijthoff and Noordhoff, Alphen aan den Rijn, The Netherlands, 1978, pp. 41-60.

[41] G. E. Kokolakis, "Bayesian Classification and Classification Performance for Independent Distributions," *IEEE Transactions on Information Theory*, vol. 27, no. 4, July 1981, pp. 500-502.

[42] R. E. Krichevsky and V. K. Trofimov, "The Performance of Universal Encoding," *IEEE Transactions on Information Theory*, vol. 27, no. 2, March 1981, pp. 199-207.

[43] Q. Li and D. W. Tufts, "Principal Feature Classification," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, January 1997, pp. 155-160.

[44] S. R. Kulkarni and O. Zeitouni, "A General Classification Rule for Probability Measures," *The Annals of Statistics*, vol. 23, no. 4, 1995, pp. 1393-1407.

[45] J. J. Martin, *Bayesian Decision Problems and Markov Chains*, John Wiley & Sons, Inc., New York, 1967.

[46] N. Merhav and Y. Ephraim, "A Bayesian classification approach with application to speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 39, no. 10, October 1991, pp. 2157-2166.

[47] N. Merhav and J. Ziv, "Estimating with Partial Statistics the Parameters of Ergodic Finite Markov Sources," *IEEE Transactions on Information Theory*, vol. 35, no. 2, March 1989, pp. 326-334.

[48] N. Merhav and J. Ziv, "A Bayesian Approach for Classification of Markov Sources," *IEEE Transactions Information Theory*, vol. 37, no. 4, July 1991, pp. 1067-1071.

[49] D. Middleton and R. Esposito, "Simultaneous Optimum Detection and Estimation of Signals in Noise," *IEEE Transactions on Information Theory*, vol. 14, no. 3, May 1968, pp. 434-444.

[50] G. F. Hughes, "Variance Comparisons for Unbiased Estimators of Probability of Correct Classification," *IEEE Transactions on Information Theory*, vol. 22, 1976, pp. 102-105.

[51] J. E. Mosimann, "On the Compound Multinomial Distribution, the Multivariate Beta-Distribution, and Correlation's Among Proportions," *Biometrika*, vol. 49, no. 1, 1962, pp. 65-82.

[52] A. Nádas, "Optimal Solution of a Training Problem in Speech Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 1, February 1985, pp. 326-329.

[53] R. E. Neapolitan, *Probabilistic Reasoning in Expert Systems*, John Wiley & Sons, Inc., New York, 1990.

[54] K. L. Oehler and R. M. Gray, "Combining Image Compression and Classification Using Vector Quantization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, May 1995, pp. 461-473.

[55] L. I. Pettit and K. D. S. Young, "Measuring the Effect of Observations on Bayes Factors," *Biometrika*, vol. 77, no. 3, 1990, pp. 455-466.

[56] M. Raghavachari, "Limiting Distributions of Kolmogorov-Smirnov Type Statistics Under the Alternative," *The Annals of Statistics*, vol. 1, no. 1, 1973, pp. 67-73.

[57] S. J. Raudys and A. K. Jain, "Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, March 1991, pp. 252-264.

[58] A. Sankar, L. Neumeyer, and M. Weintrub, "An Experimental Study of Acoustic Adaptation Algorithms," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 1996, pp. 713-716.

[59] B. M. Shahshahani and D. A. Landgrebe, "The Effect of Unlabeled Samples in Reducing the Small Sample Size Problem and Mitigating the Hughes Phenomenon," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 5, September 1904, pp. 1087-1095.

[60] M. Sobel and V. R. R. Uppuluri, "Sparse and Crowded Cells and Dirichlet Distributions," *The Annals of Statistics*, vol. 2, no. 5, 1974, pp. 977-987.

[61] J. M. Van Campenhout, "On the Peaking of the Hughes Mean Recognition Accuracy: The Resolution of an Apparent Paradox," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, no. 5, May 1978, pp. 390-395.

[62] W. G. Waller and A. K. Jain, "On the Monotonicity of the Performance of Bayesian Classifiers," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 392-394.

[63] H. Wang, D. Bell, and F. Murtagh, "Axiomatic Approach to Feature Subset Selection Based on Relevance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 3, March 1999, pp. 271-277.

[64] A. D. Wyner and J. Ziv, "Classification with Finite Memory," *IEEE Transactions on Information Theory*, vol. 42, no. 2, March 1996, pp. 337-347.

[65] Q. Xie, C. A. Laszlo, and R. K. Ward, "Vector Quantization Technique for Nonparametric Classifier Design," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 12, December 1993, pp. 1326-1329.

[66] X. Yao and Y. Liu, "A New Evolutionary System for Evolving Artificial Neural Networks," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, May 1997, pp. 694-713.

[67] J. Ziv, "On Classification with Empirically Observed Statistics and Universal Data Compression," *IEEE Transactions on Information Theory*, vol. 34, no. 2, March 1988, pp. 278-286.

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
Defense Technical Information Center	2